SPACE-TIME BLOCK CODES WITH LOW MAXIMUM-LIKELIHOOD DECODING COMPLEXITY

A Thesis
Presented to
The Academic Faculty

by

Mohanned O. Sinnokrot

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

Georgia Institute of Technology
December 2009

SPACE-TIME BLOCK CODES WITH LOW MAXIMUM-LIKELIHOOD DECODING COMPLEXITY

Approved by:

Professor John R. Barry, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Vijay Madisetti, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Ye (Geoffrey) Li
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Gorden Stuber
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Xiaoli Ma
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Alfred Andrew
School of Mathematics
Georgia Institute of Technology

Date Approved: 10 November 2009

To my father Omar and my mother Aisha.

To my wife Ola and son Motaz.

To my brothers Misbah, Mutasem, Ramzy and Mahdy.

To my father-in-law Abdul Rahman and my mother-in-law Ahida.

ACKNOWLEDGEMENTS

I want to thank my advisors Dr. John Barry and Dr. Vijay Madisetti. Dr. Barry has always given me valuable feedback and asked critical questions that helped me fill the gaps in my work. He always strived to separate the details from the main concepts and simplify the presentation of seemingly complex material. Dr. Madisetti has given the freedom to pursue research problems that interested me. He also gave me the chance to work on different projects that made me a more well-rounded engineer and taught me the value of networking.

I would like to specially thank Dr. Jim McClellan for being a great role model. I had the pleasure of working with him as a teaching assistant in ECE 2025, which is Georgia Tech's introductory signal processing class. I hope to emulate him in his dedication to research and devotion to teaching. Finally, I would like to thank him for being patient, understanding, and simply nice.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGEMENTS iv

LIST OF TABLES viii

LIST OF FIGURES ix

SUMMARY xi

1 INTRODUCTION 1

1.1 Research Objective and Contributions 7

1.2 Organization of the Thesis 10

2 SPACE-TIME BLOCK CODES 12

2.1 System and Channel Model 13

2.2 Design Criteria of Space-Time Codes 18

2.3 Decoding Complexity 23

2.4 Survey of Space-Time Block Codes 25

2.4.1 Orthogonal Space-Time Block Codes 26

2.4.2 Diagonal Algebraic Space-Time Block Codes 27

2.4.3 Quasiorthogonal Space-Time Block Codes 29

2.4.4 Single-Symbol Decodable Space-Time Block Codes 31

2.4.5 Semi-Orthogonal Space-Time Block Codes 33

2.4.6 Threaded Algebraic Space-Time Block Codes 34

2.4.7 Perfect Space-Time Block Codes 36

2.5 Conclusions 37

3 A UNIFIED FRAMEWORK FOR DETERMINING WORST-CASE MAXIMUM-LIKELIHOOD DECODING COMPLEXITY 38

3.1 The QR Decomposition of the Effective Channel Matrix 39

3.2 The Sphere Decoder 40

3.3 Key Properties for Reduced-Complexity Decoding 48

3.4 The **R** Matrix in the QR Decomposition of the Effective Channel Matrix for Space-Time Block Codes 55

3.4.1 Orthogonal Space-Time Block Codes . . . 56
3.4.2 Diagonal Algebraic Space-Time Block Codes . . . 57
3.4.3 Quasiorthogonal Space-Time Block Codes . . . 58
3.4.4 Single-Symbol Decodable Space-Time Block Codes . . . 60
3.4.5 Semi-Orthogonal Algebraic Space-Time Block Codes . . . 61
3.4.6 Threaded Algebraic Space-Time Block Codes . . . 63
3.4.7 Perfect Space-Time Block Codes . . . 65
3.5 Summary of Previous Space-Time Block Codes . . . 66
3.6 Conclusions . . . 67
4 A FAST DECODING ALGORITHM FOR THE GOLDEN CODE . . . 69
4.1 The Golden Code is Fast Decodable . . . 70
4.1.1 The Golden Code Induces Structure in Effective Channel . . . 70
4.1.2 The Golden Code is Fast Decodable . . . 72
4.2 A Fast ML Decoder With Low Average Complexity . . . 74
4.3 Golden Code Variations . . . 78
4.4 Numerical Results . . . 79
4.5 Conclusions . . . 81
5 THE ASYMMETRIC GOLDEN CODE . . . 82
5.1 A Unified Framework for the Encoding of High-Rate Space-Time Codes for the Two-Input Two-Output Channel . . . 83
5.2 The Asymmetric Golden Code . . . 85
5.2.1 The Asymmetric Golden Code Encoder . . . 85
5.2.2 The Effective Channel Matrix and its Key Properties . . . 86
5.2.3 The Asymmetric Golden Code is Fast-Decodable . . . 88
5.3 Fast ML Decoding with Low Average Decoding Complexity . . . 90
5.4 Numerical Results . . . 94
5.5 Conclusions . . . 96
6 THE EMBEDDED ORTHOGONAL SPACE-TIME BLOCK CODES . . . 97
6.1 The Embedded Orthogonal Space-Time Block Codes . . . 98
6.2 Fast ML Decoding of the Embedded Orthogonal Space-Time Codes . . . 107

6.2.1 The Effective Channel Matrix of the Embedded Orthogonal Space-Time Block Code . . . 109

6.2.2 Properties of The Effective Channel Matrix of the Embedded Orthogonal Space-Time Block Code . . . 116

6.2.3 Efficient ML Decoding of the Embedded Orthogonal Space-Time Codes . . . 118

6.3 Numerical Results . . . 124

6.4 Conclusions . . . 138

7 CONCLUSIONS AND FUTURE WORK . . . 141

7.1 Contributions . . . 141

7.2 Future Work . . . 143

APPENDIX A OPTIMUM 4, 16 AND 64-POINT HEXAGONAL CONSTELLATIONS . . . 145

APPENDIX B PROOF OF THE KEY PROPERTY FOR FAST DECODING OF THE GOLDEN CODE . . . 146

APPENDIX C OPTIMAL ASYMMETRY COEFFICIENT $\mathcal{K}$ . . . 148

APPENDIX D PROOF OF THE KEY PROPERTY FOR FAST DECODING OF THE ASYMMETRIC GOLDEN CODE . . . 152

APPENDIX E PROOF OF THE SEPARABILITY OF EMBEDDED ORTHOGONAL SPACE-TIME BLOCK CODE FOR $\mathcal{R} = \mathcal{R}_1$ . . . 154

APPENDIX F PROOF THAT THE EMBEDDED ORTHOGONAL SPACE-TIME BLOCK CODE IS NOT SEPARABLE FOR $\mathcal{R} > \mathcal{R}_1$ . . . 158

REFERENCES . . . 168

VITA . . . 168

LIST OF TABLES

2.1 Maximum Rate and Minimum Decoding Delay of Complex Orthogonal Space-Time Block Codes. 28

3.1 The **R** Matrix and its Relationship to Worst-Case Decoding Complexity. . . 51

3.2 Comparison of Different Space-Time Block Codes in Terms of Rate, Delay, Number of Real Symbols Detected Per Group, and Decoding Complexity. . 67

6.1 Design Parameters for Embedded Orthogonal Space-Time Block Codes. . . 108

6.2 Worst-Case ML Decoding Complexity for Several Space-Time Block Codes. 119

6.3 Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 4$, $N = 1$, and $\mathcal{R} = 1$. Average Complexity and SNR Penalty are Relative to 10.7, 14.7 and 19.05 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively. 127

6.4 Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 6$, $N = 1$, and $\mathcal{R} = 1$. Average Complexity and SNR Penalty are Relative to 9.3, 13.3 and 17.8 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively. 129

6.5 Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 4$, $N = 2$, and $\mathcal{R} = 1.5$. Average Complexity and SNR Penalty are Relative to 8, 12.3 and 16.8 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively. 132

6.6 Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 6$, $N = 2$, and $\mathcal{R} = 1.5$. Average Complexity and SNR Penalty are Relative to 7.1, 11.2 and 15.7 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively. 134

6.7 Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 4$, $N = 2$, and $\mathcal{R} = 2$. Average Complexity and SNR Penalty are Relative to 9.5, 14.2 and 19 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively. 136

6.8 Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 6$, $N = 2$, and $\mathcal{R} = 2$. Average Complexity and SNR Penalty are Relative to 8.7, 13.1 and 17.9 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively. 136

LIST OF FIGURES

1.1 Wireless links with transmit and receive diversity. 3

2.1 Bit-error rate performance for circulant (+), diagonal algebraic (◇) and Alamouti space-time (□) block codes. 22

2.2 Encoding architecture of single-symbol decodable space-time block codes. . 32

3.1 Classical Gram-Schmidt. 39

3.2 Three level tree search with $q = 4$. 42

3.3 Example of three level tree search with $q = 4$ and computed branch metrics. 45

3.4 Sphere decoder with $K = 3$ tree levels. 46

4.1 The structure of the proposed detection tree and its branch metrics. The cost function for the leaf node is the sum of the branch metrics, $P(\mathbf{x}) = P_1 + P_2 + P_3 + P_4$. 75

4.2 Pseudocode of a fast ML decoder for the golden code. 76

4.3 Average decoding complexity versus SNR for golden code with 64-QAM. . . 80

5.1 The structure of the proposed detection tree and its branch metrics for the asymmetric golden code. The cost function for the leaf node is the sum of the branch metrics, $P(\mathbf{x}) = P_1^R + P_1^I + P_2^R + P_2^I + P_3 + P_4$. 92

5.2 Pseudocode of a fast ML decoder for the asymmetric golden code in quasistatic fading. 93

5.3 Performance comparison for the two-input two-output channel. 95

5.4 Complexity and performance as a function of normalized Doppler $f_D T$. . . 96

6.1 Real-valued sphere decoder with K tree levels. 120

6.2 Pseudocode of a fast ML decoder for the embedded orthogonal space-time block codes for $\mathcal{R} = \mathcal{R}_1$. 121

6.3 Pseudocode of a fast ML decoder for the embedded orthogonal space-time block codes for $\mathcal{R} > \mathcal{R}_1$. 123

6.4 Performance and complexity comparison for $M = 4$, $\mathcal{R} = 1$ and $N = 1$ for quasiorthogonal (△), TAST (+), perfect (◇) and embedded orthogonal space-time (□) block codes. 126

6.5 Performance and complexity comparison for $M = 6$, $\mathcal{R} = 1$ and $N = 1$ for quasiorthogonal (△), TAST (+), perfect (◇) and embedded orthogonal space-time (□) block codes. 128

6.6 Performance and complexity comparison for $M = 4$, $\mathcal{R} = 1.5$ and $N = 2$ for TAST (+), perfect (◇), embedded orthogonal space-time (□) block code and Ismail *et al.* space-time [66] (∗) block code. 131

6.7 Performance and complexity comparison for $M = 6$, $\mathcal{R} = 1.5$ and $N = 2$ for TAST (+), perfect ($\diamond$), $\boldsymbol{\mathcal{G}}(2,1)$ embedded orthogonal ($\square$) and $\boldsymbol{\mathcal{G}}(4,\frac{3}{4})$ embedded orthogonal ($\triangledown$) block codes. 133

6.8 Performance and complexity comparison for $M = 4$, $\mathcal{R} = 2$ and $N = 2$ for TAST (+), perfect ($\diamond$) and embedded orthogonal space-time ($\square$) codes. . . 135

6.9 Performance and complexity comparison for $M = 6$, $\mathcal{R} = 2$ and $N = 2$ for TAST (+), perfect ($\diamond$) and embedded orthogonal space-time ($\square$) codes. . . 137

6.10 Performance comparison for $M = 6$, $\mathcal{R} = 3$ and $N = 3$ for TAST (+), perfect ($\diamond$) and embedded orthogonal space-time ($\square$) block codes. 139

A.1 Optimum hexagonal constellations for $q = 4$, $q = 16$ and $q = 64$. 145

SUMMARY

Multipath fading has been long viewed as an impairment in wireless communication systems that limits the reliability and data rate of the communication link. By deploying multiple antennas at the transmitter and receiver, multipath fading can be turned into an advantage, allowing for greater reliability and higher data rates than would otherwise be possible. Furthermore, the rate and reliability benefits can be achieved without extra cost of bandwidth, making multiple antenna technology a cornerstone of current and future wireless systems.

The potential benefits of multiple antenna systems can be harnessed through the use of *space-time coding*. In space-time coding, information symbols are coded across two dimensions, the spatial dimension, which corresponds to the multiple antennas at the transmitter, and time dimension, which corresponds to the multiple signaling intervals. In this thesis, we focus on *linear space-time block codes*, in which the information symbols are linearly combined to form a two-dimensional code matrix, wherein the rows of the matrix correspond to transmission across multiple intervals, and the columns of the matrix correspond to transmission from different antennas.

In this thesis, we consider the problem of designing space-time block codes that have low maximum-likelihood (ML) decoding complexity. We first present a unified framework for determining the worst-case ML decoding complexity of space-time block codes. We use this framework to not only determine the worst-case ML decoding complexity of our own constructions, but also to show that some popular constructions of space-time block codes have lower ML decoding complexity than was previously known. Specifically, we show that the *golden code*, which harnesses both the reliability and rate benefits of the two-input two-output channel, has a worst-case ML decoding complexity that is substantially lower than that of an exhaustive-search decoder.

Recognizing the practical importance of the two transmit and two receive antenna system, we propose the *asymmetric golden code*, which is designed specifically for low ML decoding complexity. Unlike some previous constructions, which lose their reduced complexity decoding when the channel varies during the transmission period of the code matrix, the asymmetric golden code maintains its low decoding complexity regardless of channel variability. The asymmetric golden code has the lowest decoding complexity compared to previous constructions of space-time codes, regardless of whether the channel varies with time.

Space-time codes that layer or multiplex rate-one space-time codes to achieve arbitrary rates ranging from one to maximal rate have been proposed in literature. Two of the most important constructions are the threaded algebraic and perfect space-time codes. These codes, however, suffer from high decoding complexity and worse bit-error-rate performance when compared to other space-time codes that have been proposed for a particular rate and a particular number of transmit and receive antennas. In this research, we propose the *embedded orthogonal space-time codes*, which is a family of codes for an arbitrary number of antennas, and for any rate up to half the number of antennas. The family of embedded orthogonal space-time codes is the first general framework for the construction of space-time codes with low-complexity decoding, not only for rate one, but for any rate up to half the number of transmit antennas. Simulation results for up to six transmit antennas show that the embedded orthogonal space-time block codes are simultaneously lower in complexity and lower in error probability when compared to some of the most important constructions of space-time block codes with the same number of antennas and the same rate larger than one.

Having considered the design of space-time block codes with low ML decoding complexity on the transmitter side, we also develop efficient algorithms for ML decoding for the golden code, the asymmetric golden code and the embedded orthogonal space-time block codes on the receiver side. Simulations of the bit-error rate performance and decoding complexity of the asymmetric golden code and embedded orthogonal codes will be used to demonstrate their attractive performance-complexity tradeoff.

CHAPTER 1

INTRODUCTION

In the last decade, there has been a dramatic increase in the demand for higher data rates in cellular networks, wireless local area networks and high-definition audio and video broadcasting services. Providing wireless access to the Internet and multimedia services requires an increase in data rates that is orders of magnitude beyond the capability of today's technology. One of the most significant and promising advances in digital communications that can meet the demand for higher data rates is the use of multiple antennas at the transmitter and receiver. Deploying multiple antennas at the transmitter and receiver creates a multiple-input multiple-output (MIMO) channel that not only offers higher transmission rates, but it can also improve the system's reliability and robustness to noise compared to single antenna systems.

Signal transmission over the wireless channel suffers not only from additive noise, but also from *multipath fading*. Specifically, a transmitted radio signal propagates through multiple paths, due to scattering and reflections from different objects in the environment, before it reaches the receive antenna. At the receiver, the multipath signals combine, and depending on the amplitude, delay and phase of each of the multipath signals, the result could be that the signals cancel each other completely or significantly attenuate the combined signal. Due to multipath fading, the received signal can be significantly attenuated that the receiver cannot correctly detect the transmitted signal.

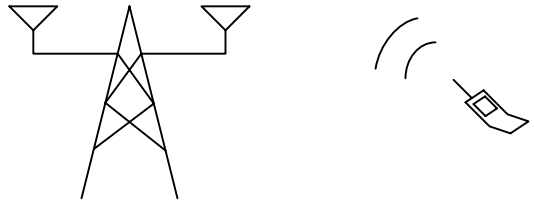
One way to overcome the problem of multipath fading is *diversity*. The basic concept of diversity is to transmit the same information symbols over multiple channels that are fading independently. This way, if one of the channels is in a deep fade, the receiver can still recover the transmitted signal if one of the other channels is in a good enough state to allow for reliable detection. Diversity can be harnessed along the time, frequency and spatial dimensions, wherein the same information is transmitted at different time instances,

transmitted in different frequency bands, or transmitted or received from different points in space, respectively.

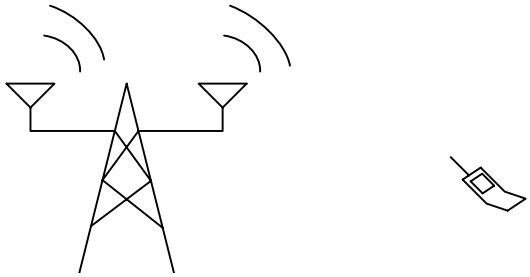
Space or spatial diversity can be exploited by using multiple antennas that are sufficiently separated at the receiver such that the signal received at each antenna undergoes independent fading. Many modern day communication systems use *receive diversity* at the base station. For example, the base stations in the global system for mobile communications (GSM) typically have two receive antennas [1]. Receive diversity at the base station improves the quality of the communication link from the mobile unit to the base station (uplink).

In the early 1990's, researchers realized that the spatial diversity benefits of receive antennas can also be harnessed at the transmitter by using multiple antennas to achieve *transmit diversity*. Since then, several transmit diversity techniques have been developed with early development efforts focusing on using the antennas at the base station, which were used for receive diversity, to harness transmit diversity. These techniques attracted a lot of attention because they improved the quality of the link from the base station to the mobile unit (downlink) without incurring the additional cost of antennas at the mobile unit. Combined with receive diversity at the base station, the quality of both the uplink and downlink is improved. Wireless links with receive and transmit diversity enabled by multiple antennas at the base station are shown in Figure 1.1(a) and Figure 1.1(b), respectively. By deploying multiple antennas at the transmitter and receiver, we can harness transmit and receive diversity at the base station and the mobile unit as shown in Figure 1.1(c).

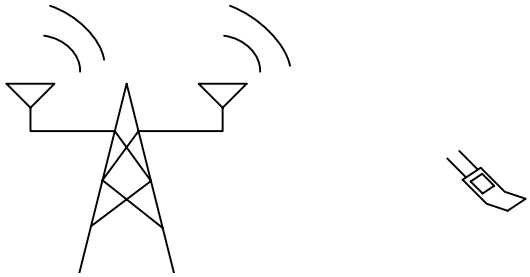
Early research efforts focused on harnessing the diversity benefit of the MIMO channel. An additional benefit of the MIMO channel in Figure 1.1(c) is that we can increase the throughput of the communication link, and do so at both ends of the link. The seminal work in [2] showed that the capacity or throughput of the channel can be significantly increased by using multiple antennas at both the transmitter and receiver. Under certain conditions, multiple independent data streams can be sent or multiplexed from the transmit antenna array and be decoded reliably at the receiver. The prospect of improving the reliability and throughput of the wireless communication link - at no cost of extra spectrum - is largely



(a) Receive diversity.



(b) Transmit diversity.



(c) Transmit and receive diversity in a MIMO channel.

Figure 1.1: Wireless links with transmit and receive diversity.

responsible for the incorporation of MIMO technology into current and future commercial wireless products and standards such as CDMA-2000, W-CDMA, Long Term Evolution (LTE), IEEE 802.11n (WiFi), and IEEE 802.16e (WiMAX).

The use of multiple antennas to increase the throughput of the communication channel or increase its reliability has been quantified by defining three related terms: rate, multiplexing gain and diversity gain. The rate of a communication system is the number of transmitted symbols per signaling interval. The multiplexing and diversity gains define how fast the rate increases or how fast the error probability decreases, respectively, with increase in signal-to-noise ratio (SNR) in the high SNR region. For any number of antennas, there is a continuous tradeoff between the diversity and multiplexing gains [3]. The diversity gain and code rate, however, can be simultaneously maximized [4][5].

A key idea in multiple antenna systems is *space-time coding*, in which the time dimension inherent in digital communications is complemented by the spatial dimension inherent in the use of multiple antennas. A key benefit of space-time coding is the ability to turn multipath propagation, an impairment in single antenna wireless communications, into a benefit for multiple antenna systems by taking advantage of the random fading in increasing the transmission rate of the communication link as well as increase its reliability. As a result of the two different benefits of MIMO channels, early transmission schemes typically fell into two categories: diversity gain maximization schemes or rate maximization schemes. However, recent work in space-time code design has shifted away from increasing the diversity gain or transmission rate alone to increasing both simultaneously.

Achieving higher rates and improved reliability in multiple antenna systems comes with a cost. Although there is no additional cost of bandwidth, there is a significant increase in complexity compared to single antenna systems. Specifically, a maximum-likelihood decoder for the transmitted signal can have a worst-case complexity that grows exponentially with the number of transmitted symbols.

Recognizing the importance of decoding complexity in practical systems, this thesis designs space-time block codes that not only harness both the rate and reliability benefits

of MIMO systems, but do so with low maximum-likelihood decoding complexity. A by-product of the design of such space-time codes is a unified framework for determining the worst-case ML decoding complexity of space-time block codes. We also address the high decoding complexity on the receiver side by presenting novel maximum-likelihood decoders with low average complexity for the practically important configuration of two transmit and two receive antennas.

We mentioned that spatial diversity can be exploited using space-time coding. Time diversity can be exploited using *error-control coding* and *interleaving*. Error-control coding introduces redundancy in the transmitted data, and interleaving spreads the redundant data symbols so that they experience independent fading. In situations where the channel varies slowly with time, we might not be able to exploit time diversity because the interleaver length might become too prohibitive for practical implementation. Specifically, a long interleaver might introduce an unacceptable amount of latency into the system. A similar situation occurs in the case of frequency diversity where the channel undergoes flat fading such that the interleaved redundant data symbols across different frequency bands do not experience independent fading. Because time and frequency diversity cannot always be exploited by error-control coding, space-time coding can complement error-control coding. Specifically, space-time coding complements error-control coding by encoding the redundant and interleaved data symbols across time and space.

Digital communication systems are often partitioned into different modules or elements, wherein error-control coding and space-time coding constitute different elements. The design and decoding of space-time block codes and error-control codes has followed this traditional partitioning such that space-time codes are designed independently of error-control codes and vice versa. In this thesis, we will only focus on the design and decoding of space-time block codes. We do so for the sole purpose of being able to compare the complexity and error-rate performance of space-time codes fairly, irrespective of a particular choice of an error-control code. We emphasize, however, that space-time coding complements error-control coding in practical systems.

An important consideration in practical systems is the availability of channel state

information (CSI) at the transmitter and receiver. Throughout this research, we will assume that the receiver has perfect knowledge of the channel, but the transmitter does not have any knowledge of it. Such system is known as *open-loop* system since there is no feedback path from the receiver to the transmitter where CSI can be communicated. Systems where CSI is also available at the transmitter are known as *closed-loop* systems. In both systems, the channel can be estimated at the receiver using a preamble or pilot symbols, which are known signals at both the transmitter and receiver. The receiver can use these known signals to estimate the channel and refine or adapt the channel estimate. In a closed-loop system, the CSI is then fed back to the transmitter.

In closed-loop systems, space-time processing known as *eigenbeamforming* is used rather than space-time block coding. Eigenbeamforming converts the MIMO channel into a bank of independent scalar channels. Furthermore, eigenbeamforming is optimal in the sense of achieving the capacity of the MIMO channel and achieving full diversity gain [6].

One of the main challenges in eigenbeamforming is how to obtain channel knowledge at the transmitter. Most current wireless standards allocate a feedback channel to transmit CSI. Furthermore, the CSI is usually *quantized* to reduce the transmission load required to feedback the CSI to the transmitter. An important issue that arises in closed-loop systems is CSI mismatch. Specifically, CSI mismatch between the transmitter and receiver due to either long feedback delay or due to a rapidly time-varying channel results in significant performance loss in eigenbeamforming. Due to their inherently different costs, both open-loop and closed-loop techniques are important in practical wireless systems. In fact, most current and future wireless systems use both open-loop and closed-loop MIMO techniques. For example, the next generation WiFi (802.11n) and WiMAX (802.16e) standards use both open-loop techniques in the form of space-time coding and closed-loop techniques in the form of eigenbeamforming.

In the remainder of this chapter, we highlight our contributions on both the transmitter side and the receiver side. Finally, we outline the organization of the remainder of the thesis.

1.1 Research Objective and Contributions

Mobile communication systems having two transmit antennas and two receive antennas are of great practical importance for two main reasons: First, a transmitter with two antennas can communicate at a higher data rate and more reliably than a transmitter with a single antenna. Second, power and size constraints prevent many devices, especially in mobile applications, from having more than two antennas. Several space-time block codes have been proposed specifically for this practically important configuration. Arguably, the most important of these codes is the golden code [7][8]. The golden code has many advantages: it is full-rate, it is fully diverse and in terms of the SNR required to achieve a target error probability, it performs better than previously reported codes. Furthermore, the coding gain of the golden code is independent of the alphabet size, which ensures that the golden code achieves the full diversity multiplexing frontier of Zheng and Tse [3][9], and which makes it compatible with adaptive modulation. For these reasons, the golden code has been incorporated into the 802.16e WiMAX standard [10].

The golden code applied to a system with two receive antennas leads to an *effective* four-input four-output channel that maps each block of four q-ary information symbols to a vector of four complex-valued received samples [11]. An exhaustive search maximum-likelihood (ML) decoder would consider each of the q^4 possible input vectors in turn and choose as its decision the one that best represents the channel output in a minimum-distance sense. Therefore, the complexity of such an exhaustive search ML decoder is proportional to q^4. Although significant reductions in *average* complexity are possible by adopting a tree-based ML decoder, the *worst-case* complexity of a sphere decoder is generally no better than that of an exhaustive search.

The perception that ML decoding of the golden code has high complexity has had two effects: First, it has motivated a search for suboptimal decoders for the golden code with reduced complexity and near-ML performance [12]-[15]. Second, it has motivated a search for lower-complexity alternatives that perform almost as well as the golden code [16]-[18].

A drawback to the use of suboptimal decoders is the performance loss compared to the maximum-likelihood decoder. Furthermore, the suboptimal decoders in [12][13] do not

exploit the special properties of the effective channel matrix induced by the golden code to potentially reduce the decoding complexity. Specifically, the decoders in [12][13] treat the golden code decoding problem in the same manner as decoding of a spatial multiplexing system with four transmit antennas, transmitting four complex information symbols per signaling interval. On the other hand, the suboptimal detectors in [14][15] do exploit the special structure of the effective channel matrix in reducing the worst-case decoding complexity. However, their average decoding complexity is the same as the worst-case complexity.

A drawback to the alternative constructions of the golden code in [16]-[18] is that they do not offer any reduction in decoding complexity when the channel varies with time, a common occurrence in wireless applications with high mobility. In particular, device mobility results in channels that could vary quickly between quasistatic fading and rapid fading, depending on the mobile speed or Doppler frequency.

We address both of these drawbacks in this research. First, we prove that the golden code with q-ary quadrature amplitude modulation (QAM) alphabet is *fast decodable*, by which we mean that ML decoding is possible with a worst-case complexity of only $\mathcal{O}(q^{2.5})$. The golden code is fast decodable regardless of whether the channel varies with time. We also present an efficient implementation of a fast decoder that has low average complexity. Second, we propose the *asymmetric golden code*, in which one layer of the golden code is scaled with respect to the other. Similar to the golden code, we prove that the asymmetric golden code has a worst-case decoding complexity of $\mathcal{O}(q^{2.5})$, regardless of whether the channel is time varying. For the special case of a slowly varying channel where the channel remains constant for the duration of a codeword, we prove that its worst-case decoding complexity is even smaller, namely $\mathcal{O}(q^2)$. Furthermore, we present an efficient implementation of a fast decoder for the asymmetric golden code.

Having discussed our contributions for the two-input two-output channel, we shift our focus to the design of space-time block codes with low maximum likelihood decoding complexity for any number of antennas. The design or choice of a space-time code depends strongly on the size of the receiver antenna array relative to the size of the transmitter

antenna array. At one extreme, when the number of receive antennas is equal or greater than the number of transmit antennas, good candidates include the threaded algebraic space-time (TAST) [4][5] codes and the perfect space-time codes [19][20], both of which achieve maximum diversity order and full transmission rate. In fact, the golden code that we mentioned earlier is an example of a perfect space-time code for the two-input two-output channel. At the other extreme, when there is only a single receive antenna, good candidates include the Alamouti code [21], the quasiorthogonal space-time block codes [22] and the semi-orthogonal algebraic space-time (SAST) block codes [23].

There are two main approaches to the construction of *high-rate* codes, by which we mean space-time block codes that have a transmission rate higher than one symbol per signaling interval. The first approach is puncturing, in which a maximal-rate space-time code is punctured to obtain the high-rate code. For example, TAST and perfect space-time codes for four transmit antennas of rate $\mathcal{R} \in \{1, 2, 3, 4\}$ are easily obtained by puncturing $4 - \mathcal{R}$ threads of the rate-4 code. The second approach is multiplexing, in which lower rate space-time codes are multiplexed or combined to form the high-rate code. For example, the rate-two space-time code in [24] for the four-input two-output channel is constructed as a combination of two quasiorthogonal codewords.

High-rate space-time block codes constructed by puncturing or multiplexing suffer from high decoding complexity and/or worse bit-error-rate performance when compared to other space-time codes that have been proposed for a particular rate and a particular number of transmit and receive antennas. For example, the SAST [23] codes not only outperform the rate-one TAST and perfect codes in terms of the SNR required to achieve a target error rate, but they also have a lower decoding complexity. A general framework for the design of high-rate space-time codes with low decoding complexity remains elusive.

This research examines the problem of designing *high-rate* space-time codes whose rate is larger than one but less than full, appropriate in between the two extremes. In addressing the design problem, we introduce the concept of *embedding*, in which complex orthogonal designs [25][26] assume the role of complex symbols in the encoding process. Based on the

embedding concept we propose a new family of codes called *embedded-orthogonal space-time* (EOS) codes, defined for an arbitrary number of antennas and for any rate up to half the number of antennas. When compared to previously reported space-time codes, the proposed family of codes is lower in decoding complexity. Furthermore, simulation results up to six transmit antennas show that the proposed embedded orthogonal space-time codes outperform quasiorthogonal [22], diagonal algebraic [27][28], thread algebraic [4] and perfect space-time block codes [19] for all rates higher than one symbol per channel use, when performance is measured in terms of the SNR required to achieve a target error probability.

1.2 Organization of the Thesis

Chapter 2 presents background information related to space-time coding. More specifically, we present the channel model, review the design criteria and survey the different constructions of space-time block codes.

The novel contributions of this thesis are presented in the next four chapters.

- Chapter 3 presents a unified framework for determining the worst-case ML decoding complexity. This framework is used to show that the worst-case ML decoding complexity for certain families of space-time block codes is less than what is reported for an exhaustive-search ML decoder. The framework is used in subsequent chapters to determine the worst-case ML decoding complexity of the golden code, the proposed asymmetric golden code, and the proposed embedded orthogonal space-time codes.

- Chapter 4 uses the unified framework in Chapter 3 to prove that the golden code is *fast-decodable* and that it retains its fast decodability on quasistatic fading as well as rapid time-varying fading. Chapter 4 also presents an efficient ML decoding algorithm with low average decoding complexity [29][30].

- Chapter 5 presents the asymmetric golden code, which has a lower decoding complexity than the golden code on quasistatic channels with QAM alphabet and the same

decoding complexity on rapid time-varying channels. Efficient ML decoding algorithms with low average decoding complexity for quasistatic and time-varying fading are also presented [31][32].

- Chapter 6 presents the embedded-orthogonal space-time (EOS) codes, a family of codes for any number of antennas, and for any rate up to half the number of antennas. For rate larger than one, simulation results show that embedded orthogonal space-time codes are simultaneously lower in complexity and lower in error probability on quasistatic Rayleigh channels than previous constructions of space-time codes [33][34].

Chapter 7 summarizes the conclusions from this research and discusses some areas for future research.

CHAPTER 2

SPACE-TIME BLOCK CODES

Space-time coding is a method used in multiple antenna systems to not only increase the reliability of the communication link, but also increase its throughput. This is accomplished by encoding multiple streams of data across the spatial domain (i.e., antennas) and across the time domain.

The delay diversity scheme of Wittenben [35] inspired the first attempt to develop space-time codes in [36]. The key development of the space-time coding concept is due to Tarokh *et al.* in [37], in which they proposed *space-time trellis codes*. These space-time trellis codes had a high decoding complexity and required a vector Viterbi algorithm at the receiver for decoding. In addressing the high decoding complexity of space-time trellis codes, *space-time block codes* (STBC) were discovered. The first space-time block code was a remarkable scheme for two transmit antennas developed by Alamouti [21]. The Alamouti space-time block code allowed the receiver to harness the spatial diversity of the transmit antennas while maintaining simple maximum-likelihood (ML) decoding at the receiver.

The first family of space-time block codes proposed in the literature is orthogonal designs [26], in which all the transmitted symbols are decoded independently. The maximum rate of orthogonal designs, however, is less than one symbol per signaling interval for more than two antennas. The diagonal algebraic [27] and quasiorthogonal space-time block codes [38]-[43] achieved a transmission rate of one symbol per signaling interval, at the expense of an increase in decoding complexity. The semi-orthogonal algebraic space-time block codes [23] were proposed as an alternative to quasiorthogonal and diagonal algebraic space-time codes because they achieve comparable bit-error rate performance, but they require a lower decoding complexity.

Orthogonal, quasiorthogonal, diagonal algebraic and semi-orthogonal algebraic space-time block codes achieve a maximum rate of one symbol per signaling interval. In addressing

the need for higher rates, the threaded algebraic [4] and perfect space-time block codes [19][20] were proposed. These high-rate space-time codes achieve the maximum rate of the communication link.

In this chapter, we provide background information related to the problems being pursued in this research. First, we describe the MIMO communication system model and the channel model. Next, we briefly review some useful design criteria for space-time block codes in quasistatic fading channels. Then, we discuss maximum-likelihood (ML) decoding of space-time block codes and define a few terms related to ML decoding. Finally, we provide a survey of many important constructions of space-time block codes including orthogonal, quasiorthogonal, semi-orthogonal algebraic, diagonal algebraic, thread algebraic, and perfect space-time block codes.

2.1 System and Channel Model

We consider a MIMO system with M transmit antennas and N receive antennas. The transmitted codeword of a space-time block code can be expressed as a $T \times M$ matrix:

$$\mathbf{C} = \begin{bmatrix} c_1[1] & c_2[1] & \cdots & c_M[1] \\ c_1[2] & c_2[2] & \cdots & c_M[2] \\ \vdots & \vdots & \ddots & \vdots \\ c_1[T] & c_2[T] & \cdots & c_M[T] \end{bmatrix}, \tag{2.1}$$

where $c_m[t]$ denotes the symbol transmitted from antenna $m \in \{1, \cdots, M\}$ at time $t \in \{1, \cdots, T\}$. The received signal $y_n[t]$ at receive antenna $n \in \{1, \cdots, N\}$ at time t is given by:

$$y_n[t] = \sum_{m=1}^{M} h_{m,n}[t] c_m[t] + w_n[t], \tag{2.2}$$

where

- $w_n[t]$ is the complex additive white Gaussian noise at receive antenna n at time interval t with $E(|w_n[t]|^2) = N_0$.
- $h_{m,n}[t]$ is the channel coefficient between the m-th transmit antenna and n-th receive antenna at time t.

Throughout the thesis, we will assume *Rayleigh fading*, wherein the channel coefficients $h_{m,n}[t]$ are independent and identically distributed (i.i.d) complex Gaussian random variables with variance of one half. The Rayleigh fading model accurately describes a rich scattering environment with many reflectors. It is also used in a typical cellular environment even with a relatively small number of reflectors [44].

We next define a few terms related to the channel model, *delay spread*, *flat fading*, and *quasistatic* and *time-varying* fading.

Definition 2.1. Multipath *delay spread* is the difference in propagation time between the longest and shortest paths, when we only count the paths with significant energy.

Definition 2.2. A channel is *flat fading* when the delay spread of the channel is much less than the symbol period of the transmitted signal such that the frequency response of the channel is flat over the allocated bandwidth of the system.

In this thesis, we will assume flat fading. Systems with large bandwidth where the frequency response of the channel is not flat over the signal bandwidth are common in practice. The assumption of flat fading, however, is not restrictive since systems with large bandwidth can be combined with orthogonal frequency division multiplexing (OFDM) technique that effectively transforms the large bandwidth signal into multiple parallel smaller bandwidth signals such that the fading on each smaller bandwidth signal is flat.

Definition 2.3. Fading is *quasistatic* when the channel coefficient $h_{m,n}[t]$ is assumed constant over a block of T symbol periods, but it is independent from block to block. The channel fading coefficient is then independent of the time index t such that $h_{m,n}[t] = h_{m,n}$.

Definition 2.4. Fading is *time-varying* when the channel coefficient $h_{m,n}[t]$ is not constant over T symbol periods.

In this thesis, we consider both quasistatic and time-varying fading. In fact, the ML decoder for the golden code in Chapter 4 applies equally to quasistatic and time-varying channels. Furthermore, the asymmetric golden code in Chapter 5 is designed for low decoding complexity on both quasistatic and time-varying channels.

We next define what it means to be a *linear* space-time block code.

Definition 2.5. A *linear space-time block code* is a $T \times M$ matrix whose entries are complex linear combinations of K complex information symbols $x_k = x_k^R + ix_k^I$, $k \in \{1, \cdots, K\}$, and their complex conjugates, where $i = \sqrt{-1}$ and the superscripts "R" and "I" denote the real and imaginary components of the complex information symbol x_k, respectively.

Example 2.1. The Alamouti space-time block code [21] is:

$$\mathbf{C} = \begin{bmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{bmatrix}. \tag{2.3}$$

The Alamouti code is clearly linear transmitting two complex information symbols in two time intervals. Note that the Alamouti space-time code transmits both the complex information symbol and its conjugate. □

In this thesis, we restrict our attention to *linear* space-time block codes. This is not a restrictive assumption since linear space-time block codes have been shown to be optimal in terms of maximizing the mutual information between the transmit and receive signals [45]. Furthermore, linear space-time block codes are also optimal in terms of the diversity-multiplexing tradeoff of the MIMO channel [19][20]. In addition, linear space-time block codes can be decoded efficiently using linear processing algorithms such as the sphere decoding algorithm [46][47], which was proposed for the ML decoding of lattice codes and space-time block codes in [48] and [49].

The transmitted complex information symbols for a linear space-time block code are drawn from an alphabet $\mathcal{A}$, with cardinality q (i.e., $q = |\mathcal{A}|$). In particular, the complex information symbols are drawn from one of three alphabets:

- quadrature amplitude modulation (QAM) alphabet. A QAM alphabet is a subset of $\mathbb{Z}[i] \triangleq \{a + bi\}$, $a, b \in \mathbb{Z}$, where $i = \sqrt{-1}$. Specifically, a q-ary QAM alphabet is $\mathcal{A} = \{\pm 1, \pm 3, \cdots, \pm(\sqrt{q}-1)\} + i\{\pm 1, \pm 3, \cdots, \pm(\sqrt{q}-1)\}$. The real or imaginary part of the QAM alphabet defines a real alphabet known as pulse-amplitude modulation (PAM) alphabet.

- hexagonal (HEX) alphabet. A HEX alphabet is constructed from the two-dimensional hexagonal lattice $\mathbb{Z}[j] \triangleq \{a + bj\}$, $a, b \in \mathbb{Z}$, where $j = e^{i2\pi/3}$. The optimum hexagonal

constellations, in terms of minimizing the error probability for a given SNR, were studied in [50][51]. We illustrate the optimum hexagonal constellations for 4-HEX, 16-HEX, and 64-HEX in Appendix A.

- phase shift keying (PSK) alphabet. A PSK alphabet is given by $\mathcal{A} = \{e^{i2\pi p/q} : p \in \{0, 1, \cdots, q-1\}\}$.

We next present a more useful form of the system model in (2.2). We begin the discussion with an example.

Example 2.2. Let us consider the Alamouti space-time block code in Example 2.1. Assuming quasistatic fading and a single receive antenna, the received signal at the two time instances is given by

$$\begin{aligned} y_1[1] &= h_{1,1}x_1 + h_{2,1}x_2 + w_1[1], \\ y_1[2] &= -h_{1,1}x_2^* + h_{2,1}x_1^* + w_1[2]. \end{aligned} \tag{2.4}$$

It is usually desired to express the system of equations in (2.4) in matrix form as a function of *only* the information symbols, and not their conjugates. This can be done by conjugating the second received sample as follows:

$$\underbrace{\begin{bmatrix} y_1[1] \\ y_1^*[2] \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} h_{1,1} & h_{2,1} \\ h_{2,1}^* & -h_{1,1}^* \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} w_1[1] \\ w_1^*[2] \end{bmatrix}}_{\mathbf{w}}. \tag{2.5}$$

Note that the vector $\mathbf{y}$ contains either the received sample or its conjugate. Similarly, the vector $\mathbf{w}$ contains the noise sample or its conjugate. This is necessary in order to express the system as a function of the vector $\mathbf{x}$ that contains only the information symbols, but not their conjugates. □

Having discussed the specific case of the Alamouti space-time block code in Example 2.2, we now consider the general system model in (2.2). In order to facilitate the use of efficient decoding algorithms, the system model in (2.2) can be expressed in the form

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}, \tag{2.6}$$

where

- $\mathbf{y} = [\tilde{y}_1[1], \tilde{y}_1[2], \cdots, \tilde{y}_1[T-1], \tilde{y}_1[T], \cdots, \tilde{y}_N[1], \tilde{y}_N[2], \cdots, \tilde{y}_N[T-1], \tilde{y}_N[T]]^\top$ is the $TN \times 1$ complex vector of received samples or their conjugates at all time intervals, from all receive antennas
- $\tilde{y}_n[t]$ is the received sample $y_n[t]$ or its conjugate $y_n^*[t]$
- $\mathbf{H}$ is a $TN \times K$ matrix, termed the *complex-valued effective channel matrix*
- $\mathbf{x} = [x_1, x_2, \cdots, x_K]^\top$ is $K \times 1$ vector of complex information symbols
- $\mathbf{w} = [\tilde{w}_1[1], \tilde{w}_1[2], \cdots, \tilde{w}_1[T-1], \tilde{w}_1[T], \cdots, \tilde{w}_N[1], \tilde{w}_N[2], \cdots, \tilde{w}_N[T-1], \tilde{w}_N[T]]^\top$ is the $TN \times 1$ vector of noise samples or their conjugates at all time intervals, from all receive antennas.

The system model in (2.2) is a linear system of equations in the complex information symbols *and* their conjugates, and hence, it is not always possible to express (2.2) in matrix form as a function of only the complex information symbols as in (2.6). We overcome this problem by converting the complex-valued system in (2.6) to the equivalent real-valued system as follows

$$\check{\mathbf{y}} = \check{\mathbf{H}}\check{\mathbf{x}} + \check{\mathbf{w}}, \tag{2.7}$$

where

- $\check{\mathbf{y}} = [y_1^R[1], y_1^I[1], \cdots, y_1^R[T], y_1^I[T], \cdots, y_N^R[1], y_N^I[1], \cdots, y_N^R[T], y_N^I[T]]^\top$ is the *real* vector of size $2TN \times 1$ of received samples at all time intervals, from all receive antennas
- $\check{\mathbf{H}}$ is a $2TN \times 2K$ matrix, termed *real-valued effective channel matrix*
- $\check{\mathbf{x}} = [x_1^R, x_1^I, x_2^R, x_2^I, \cdots, x_K^R, x_K^I]^\top$ is the $2K \times 1$ vector of transmitted *real* information symbols
- $\check{\mathbf{w}} = [w_1^R[1], w_1^I[1], \cdots, w_1^R[T], w_1^I[T], \cdots, w_N^R[1], w_N^I[1], \cdots, w_N^R[T], w_N^I[T]]^\top$ is the $2TN \times 1$ *real* vector of noise samples at all time intervals, from all receive antennas.

The real-valued system model in (2.7) will be used whenever it is more convenient or simply not possible to use the complex-valued model. In particular, the real-valued

system model will be used when we discuss the decoding of orthogonal and the proposed embedded-orthogonal space-time codes. For all other space-time block codes considered in this research, it is more convenient to use the complex-valued system model in (2.6).

In Example 2.2, we discussed the complex-valued system model of the Alamouti space-time block code. We next give an example of the real-valued system model of the Alamouti code.

Example 2.3. From the definition of the Alamouti space-time block code in (2.3) in Example 2.1 and using the real-valued system model in (2.7), the received signal is given by

$$\check{\mathbf{y}} = \begin{bmatrix} y_1^R[1] \\ y_1^I[1] \\ y_1^R[2] \\ y_1^I[2] \end{bmatrix} = \begin{bmatrix} h_{1,1}^R & -h_{1,1}^I & h_{2,1}^R & -h_{2,1}^I \\ h_{1,1}^I & h_{1,1}^R & h_{2,1}^I & h_{2,1}^R \\ h_{2,1}^R & h_{2,1}^I & -h_{1,1}^R & -h_{1,1}^I \\ h_{2,1}^I & -h_{2,1}^R & -h_{1,1}^I & h_{1,1}^R \end{bmatrix} \begin{bmatrix} x_1^R \\ x_1^I \\ x_2^R \\ x_2^I \end{bmatrix} + \begin{bmatrix} w_1^R[1] \\ w_1^I[1] \\ w_1^R[2] \\ w_1^I[2] \end{bmatrix} = \check{\mathbf{H}}\check{\mathbf{x}} + \check{\mathbf{w}}. \tag{2.8}$$

□

We next define the *rate* of a space-time block code. We also define other terms used in describing the rate.

Definition 2.6. The *rate* of a $T \times M$ space-time block code transmitting K complex information symbols over T symbol periods is $\mathcal{R} = K/T$ symbols per signaling interval.

Definition 2.7. A space-time block code with M transmit is *full-rate* if $\mathcal{R} = M$.

Definition 2.8. A space-time block code with M transmit antennas is *high-rate* if $\mathcal{R} > 1$.

2.2 Design Criteria of Space-Time Codes

In this section we discuss some useful design criteria for space-time block codes on quasistatic channel fading; the rank and determinant criteria. The design criteria of space-time codes are derived based on the union bound for the codeword error probability. Assume that a codeword $\mathbf{C}$ is transmitted, wherein the columns of the codeword are normalized so that a randomly chosen column has unit energy. Assuming quasistatic Rayleigh fading, and given

perfect knowledge of the channel at the receiver, the *pairwise* error probability, which is the probability that a maximum-likelihood receiver prefers another codeword $\tilde{\mathbf{C}}$ to the actual codeword $\mathbf{C}$ is bounded by [52]

$$P_{(\mathbf{C}\to\tilde{\mathbf{C}}|\mathbf{H})} \leq \left(\frac{\Gamma\rho}{4M}\right)^{-rN}, \tag{2.9}$$

where

- r is the rank of the $M \times M$ matrix:

$$\mathbf{X} = (\mathbf{C} - \tilde{\mathbf{C}})^*(\mathbf{C} - \tilde{\mathbf{C}}), \tag{2.10}$$

- $\Gamma = (\lambda_1\lambda_2\cdots\lambda_r)^{1/r}$ is the geometric mean of the nonzero eigenvalues of the matrix $\mathbf{X}$,

- ρ is the signal-to-noise ratio (SNR) per receive antenna.

Assuming all codewords are equally likely, the union bound for the average error probability is related to the pairwise error probability as follows:

$$\begin{aligned} P_e &\leq \frac{1}{q^K}\sum_{\forall\mathbf{C}}\sum_{\substack{\forall\tilde{\mathbf{C}}\\ \tilde{\mathbf{C}}\neq\mathbf{C}}} P_{(\mathbf{C}\to\tilde{\mathbf{C}}|\mathbf{H})} \\ &\leq \left(q^K - 1\right)\max_{\mathbf{C}\neq\tilde{\mathbf{C}}} P_{(\mathbf{C}\to\tilde{\mathbf{C}}|\mathbf{H})}. \end{aligned} \tag{2.11}$$

The second inequality in (2.11) results when each pairwise error probability is replaced the worst-case value. The pairwise error probability in (2.9) along with the union bound for the average error probability in (2.11) lead us to the design criteria of space-time codes, namely the *rank criterion* and *determinant criterion* [37].

- *Rank Criterion*: To achieve maximum diversity order of MN, the matrix $\mathbf{X}$ in (2.10) must have rank M for any pair of distinct codewords $\mathbf{C}$ and $\tilde{\mathbf{C}}$.

  The rank criterion follows from (2.9) and (2.11). Specifically, the exponent $d = rN$ in (2.9) is the *diversity order* or *diversity gain* of the pairwise error probability, and it determines how fast the pairwise error probability decays, as a function of the average SNR. From (2.11), we have that the error probability will be dominated by

the worst-case pairwise error probability. Therefore, the maximum diversity order of a space-time code is MN, which is achieved when the rank of the matrix $\mathbf{X}$ is equal to M.

- *Determinant Criterion*: To further optimize performance, a space-time code that achieves diversity order MN should be chosen to maximize the asymptotic coding gain:

$$\begin{aligned} \Gamma &= \min \det(\mathbf{X})^{1/M} \\ &= \min_{\mathbf{C}-\tilde{\mathbf{C}}} \det(\mathbf{C}-\tilde{\mathbf{C}})^{*}(\mathbf{C}-\tilde{\mathbf{C}})^{1/M}. \end{aligned} \tag{2.12}$$

  The determinant criterion also follows from (2.9) and (2.11). The factor Γ in (2.9) decreases the pairwise error probability since it amplifies the SNR. Consequently, Γ can be viewed as a *pairwise coding gain*. From (2.11), the *coding gain* for a space-time code is the worst-case pairwise coding gain. Therefore, we can decrease the error rate by maximizing the coding gain in (2.12).

Both the diversity and coding gain improve system performance by decreasing the error rate, but they do so in two different ways. We discuss the difference between the coding and diversity gain by taking the logarithm of both sides of (2.11) to obtain

$$\log(P_e) \leq c - d\log(\rho), \tag{2.13}$$

where c is a constant that depends in part on the coding gain and number of transmit antennas, and where $d = rN$ is the diversity gain. From (2.13), we see that the diversity gain improves system performance by increasing the magnitude of the slope of the error rate curve, when the error rate curve is plotted on a logarithmic scale. The coding gain, on the other hand, shifts the error rate curve to the left. Furthermore, the SNR advantage due to diversity gain increases with an increase in SNR but remains constant with the coding gain at high enough SNR. We now discuss an example that highlights the differences between the diversity and coding gain in improving performance.

Example 2.4. Consider three space-time block codes for two transmit antennas, $\mathbf{C}_{circulant}$,

$\mathbf{C}_{DAST}$ and $\mathbf{C}_{Alamouti}$ given by the following code matrices:

$$\mathbf{C}_{circulant} = \begin{bmatrix} x_1 & x_2 \\ x_2 & x_1 \end{bmatrix}, \tag{2.14}$$

$$\mathbf{C}_{DAST} = \begin{bmatrix} cx_1 + sx_2 & 0 \\ 0 & -sx_1 + cx_1 \end{bmatrix}, \tag{2.15}$$

$$\mathbf{C}_{Alamouti} = \begin{bmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{bmatrix}, \tag{2.16}$$

where $c = \cos(\theta)$, $s = \sin(\theta)$, and $\theta = \frac{1}{2}\tan^{-1}(2)$. Assume that the information symbols are drawn from 4-QAM. Let us discuss each matrix in terms of the diversity and coding gain.

The matrix $\mathbf{C}_{circulant}$ is a 2×2 circulant matrix. One can easily verify that the maximum diversity gain is 1. This is because the matrix does not have rank 2 for all distinct pairs of codeword matrices. For example, the rank of $\begin{bmatrix} 1+i & 0 \\ 0 & 1+i \end{bmatrix} - \begin{bmatrix} 0 & 1+i \\ 1+i & 0 \end{bmatrix} = \begin{bmatrix} 1+i & 1+i \\ 1+i & 1+i \end{bmatrix}$ is one. We will not consider the coding gain since it is meaningful when the rank of the matrix is 2.

The matrix $\mathbf{C}_{DAST}$ is the diagonal algebraic space-time (DAST) block code for two transmit antennas. We will discuss this important family of space-time block codes later in the chapter. For the sake of this example, however, it is sufficient to note that the diagonal algebraic space-time block code achieves second-order diversity gain. Furthermore, the coding gain is $\Gamma = 3.2$.

The matrix $\mathbf{C}_{Alamouti}$ is the Alamouti space-time block code. We will also discuss the Alamouti code later in the chapter. We simply note that the Alamouti space-time block code also achieves second-order diversity gain, and its coding gain is $\Gamma = 4$.

Of course, for the case of 4-QAM, one can easily verify by exhaustive-search over all pairs of distinct codeword matrices that $\mathbf{C}_{DAST}$ and $\mathbf{C}_{Alamouti}$ achieve second order diversity, with coding gain of $\Gamma = 3.2$ and $\Gamma = 4$, respectively.

The bit-error rate performance of the three codes is given in Figure 2.1. The slope of the three error rate curves can be calculated from Figure 2.1. Specifically, one can easily

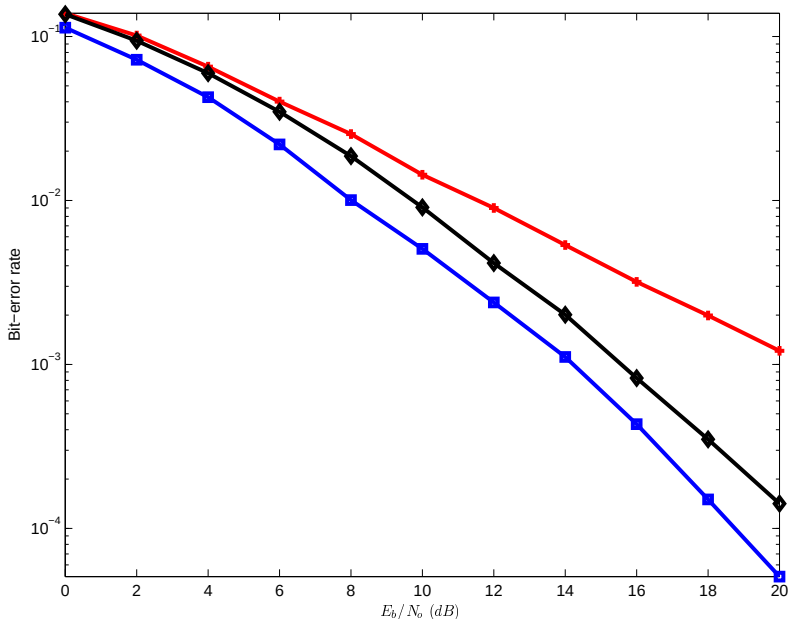


Figure 2.1: Bit-error rate performance for circulant (+), diagonal algebraic (◇) and Alamouti space-time (□) block codes.

verify that the slope of the error rate curve for the circulant, DAST and Alamouti codes is -1.05, -1.92, and -2.17; respectively, where the slope is calculated in the 14 dB to 20 dB SNR region. This is expected since the diversity gain for the circulant, DAST and Alamouti codes is 1, 2, and 2; respectively. for As can be seen from Figure 2.1, the circulant code performs worse than the diagonal algebraic and Alamouti space-time block codes, in terms of the SNR required to achieve a target error probability. This is expected since the circulant code has a diversity gain of 1. For the diagonal algebraic and Alamouti space-time block code, we see that there is a change in the slope of the error-rate curve compared to the circulant code due to the higher diversity gain. The difference in performance between the diagonal algebraic and Alamouti space-time block code is due to the fact that the Alamouti code has higher coding gain than the diagonal algebraic code, which is manifested as a shift of the entire error-rate curve more to the left. □

We next give two definitions related to the rank and determinant criteria.

Definition 2.9. A space-time block code is said to be *fully diverse* or has *full diversity* if the matrix $\mathbf{X}$ in (2.10) has rank M for any pair of distinct codewords $\mathbf{C}$ and $\tilde{\mathbf{C}}$.

Let $\mathbf{C}$ be a space-time block code encoding information symbols from a QAM or HEX alphabet without normalizing the alphabet for the average transmit energy. In other words, the average energy of the alphabet is allowed to grow as the size of the alphabet grows. For example, the average energy of 4-QAM, 16-QAM and 64-QAM is 2, 10 and 42, respectively. Then, we have the following definition related to the coding gain of a fully diverse space-time block code.

Definition 2.10. A space-time block code is said to have the *nonvanishing determinant property* if, without normalizing the codeword for the average energy of the alphabet, the coding gain for a fully diverse code is bounded away from zero as the alphabet size grows.

Designing space-time block codes that satisfy the nonvanishing determinant property is desirable since it was shown in [53] that a sufficient condition for a space-time block code to achieve the optimal diversity-multiplexing tradeoff is that it satisfies the nonvanishing determinant property.

2.3 Decoding Complexity

Assuming perfect channel state information, the maximum-likelihood decoder at the receiver chooses K complex information symbols $x_k, k \in \{1, \cdots, K\}$, that minimize the metric:

$$P(x_1, \cdots, x_K) = \sum_{t=1}^{T} \sum_{n=1}^{N} \left| y_n[t] - \sum_{m=1}^{M} h_{m,n} c_m[t] \right|^2 . \tag{2.17}$$

We next define three terms related to ML decoding.

Definition 2.11. The *decoding complexity* of an ML decoder is the number of metric computations required to reach the ML decision. The decoding complexity cannot exceed q^K metric computations, which is the *worst-case decoding complexity* achieved by an exhaustive-search ML decoder.

We next define what it means for a space-time block code to be η-group decodable. Let us first group the $2K$ real information symbols of a space-time block code into η groups such that each group has K_l real symbols, $l \in \{1, \cdots, \eta\}$. We thus have that $\sum_{l=1}^{\eta} K_l = 2K$. Furthermore, let ζ_l denote the set of indices for the l-th group. Then, we have the following definition of η-group decodable space-time block code.

Definition 2.12. A linear space-time block code is said to *η-group decodable* if the ML decoding metric in (2.17) can be decoupled into a linear sum of η independent metrics such that each metric consists of the symbols from only one group. Specifically, the cost function in (2.17) can be written as

$$\begin{aligned} P(x_1, \cdots, x_K) &= P(\check{x}_1, \cdots, \check{x}_{2K}) \\ &= \sum_{l=1}^{\eta} P_l(\check{x}_{\zeta_l(1)}, \check{x}_{\zeta_l(2)}, \cdots, \check{x}_{\zeta_l(K_l)}). \end{aligned} \tag{2.18}$$

Definition 2.13. A linear space-time block code is said to be *separable* if it is η-group decodable for any $\eta > 1$.

We next give an example to clarify these definitions.

Example 2.5. Consider three 4×4 space-time block codes $\mathbf{C_X}$, $\mathbf{C_Y}$ and $\mathbf{C_Z}$ transmitting four complex information symbols drawn from an arbitrary q-ary alphabet. In other words, we have $T = 4$, $M = 4$, and $\mathcal{R} = 1$ for all three space-time block codes. Assume that the metric function in (2.17) can be written in its simplest form as follows for the three codes:

$$\begin{aligned} \mathbf{C_X} &: P(x_1, x_2, x_3, x_4) = P_1(x_1) + P_2(x_2) + P_3(x_3) + P_4(x_4) \\ \mathbf{C_Y} &: P(x_1, x_2, x_3, x_4) = P_1(x_1, x_2) + P_2(x_3, x_4) \\ \mathbf{C_Z} &: P(x_1, x_2, x_3, x_4) = P(x_1, x_2, x_3, x_4) \end{aligned}$$

According to Definition 2.7, all three codes are not full-rate since they transmit only one symbol per signaling interval out of the maximum four symbols per signaling interval. Furthermore, the space-time block code $\mathbf{C_X}$ is separable or four-group decodable since we can separate the decoding of the transmitted symbols into four groups, each containing one complex symbol. Similarly, space-time block code $\mathbf{C_Y}$ is separable or two-group decodable since we can separate the decoding of the transmitted symbols into two groups, each containing two complex symbols. Finally, space-time block code $\mathbf{C_Z}$ is not separable since all the symbols have to be decoded jointly. □

We remark that from a practical point of view, the complexity of an algorithm is truly measured by the complexity of its hardware implementation. Hardware complexity itself

is usually measured in terms of the number of logic gates or area required to implement the algorithm, and in terms of the amount of power it dissipates. Although the worst-case ML decoding complexity is not the same as hardware complexity, it provides an accurate measure of the complexity of hardware implementation. Therefore, throughout the thesis, we will use the worst-case complexity as one measure to compare space-time block codes in terms of their decoding complexity. In Chapter 3, we will introduce another useful measure of complexity, the *average complexity*. We will defer the discussion of average complexity to Chapter 3.

2.4 Survey of Space-Time Block Codes

Having presented the channel model, and the design criteria of space-time block codes, we next provide a survey of several different constructions of space-time block codes in the literature. We review some of the most important constructions of space-time block codes, including orthogonal space-time codes, diagonal algebraic space-time codes, quasiorthogonal space-time codes, single-symbol decodable space-time codes, semi-orthogonal algebraic space-time codes, threaded algebraic space-time block codes and perfect space-time codes.

We note that all these families of space-time codes are fully diverse. We will classify these families of space-time block codes in terms of their group decodability. Although group decodability affects the worst-case ML decoding complexity, it is not the only factor in determining the worst-case decoding complexity. However, determining the group decodability of a space-time block code is still useful. For example, consider a space-time block code that is η-group decodable. If the worst-case decoding complexity of all the groups is the same, then the worst-case decoding complexity of the code is equal to the worst-case decoding complexity of any one group. We will defer the discussion of worst-case decoding complexity to Chapter 3, where we will revisit these families of space-time block codes and present a unified framework for comparing them in terms of their worst-case ML decoding complexity.

2.4.1 Orthogonal Space-Time Block Codes

An orthogonal space-time block code for M transmit antennas is a linear code that satisfies the following orthogonality property:

$$\mathbf{C}^*\mathbf{C} = \mathbf{I}_M \sum_{k=1}^{K} |x_k|^2, \tag{2.19}$$

where $\mathbf{I}_M$ is the $M \times M$ identity matrix. Orthogonal space-time block codes are an important family of linear space-time codes that achieve full diversity, while decoupling the ML detection of the transmitted symbols such that each transmitted symbol is detected separately from the other transmitted symbols. Specifically, orthogonal space-time codes are $2K$-group decodable for QAM alphabets, and K-group decodable for HEX or PSK alphabets.

The first orthogonal space-time block code is a rate-one space-time block code for two transmit antennas developed by Alamouti [21]. Tarokh *et al.* [25] constructed orthogonal space-time block codes for real and complex alphabets for arbitrary number of antennas. In particular, Tarokh *et al.* constructed real orthogonal space-time block codes with rate-one for any number of transmit antennas and complex orthogonal space-time block codes with rate 1 for two antennas, rate $\frac{3}{4}$ for three and four antennas, and rate $\frac{1}{2}$ for more than four antennas. Orthogonal space-time codes for two [21], three, and four transmit antennas [54] are given by:

$$\mathbf{C} = \begin{bmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{bmatrix}, \mathbf{C} = \begin{bmatrix} x_1 & 0 & -x_2^* \\ 0 & x_1 & -x_3 \\ x_2 & x_3^* & x_1^* \\ -x_3 & x_2^* & 0 \end{bmatrix}, \text{ and } \mathbf{C} = \begin{bmatrix} x_1 & 0 & -x_2^* & x_3^* \\ 0 & x_1 & -x_3 & -x_2 \\ x_2 & x_3^* & x_1^* & 0 \\ -x_3 & x_2^* & 0 & x_1^* \end{bmatrix}. \tag{2.20}$$

The construction of maximal rate complex orthogonal space-time block codes was studied by Liang in [26]. Liang not only determined the maximal rate achieved by complex orthogonal designs for arbitrary number of antennas, but also gave a systematic construction of maximal rate complex orthogonal space-time block codes. Furthermore, the constructed codes were also shown to be delay optimal for six transmit antennas or less. A delay-optimal orthogonal design refers to a design that achieves the maximal rate while minimizing the

code length T. Delay optimality is a desirable property because it minimizes the decoding delay at the receiver since the receiver has to receive the entire block before it starts decoding it. The maximal rate of a complex orthogonal space-time block code with M transmit antennas is given by [26]

$$\mathcal{R} = \frac{\lceil M/2 \rceil + 1}{2\lceil M/2 \rceil} \tag{2.21}$$

The minimum delay for the maximum rate complex orthogonal designs for M antennas is given by [26][55]

$$D_{min} = \frac{1}{(1+\delta)\mathcal{R}} \begin{pmatrix} M \\ \lceil M/2 \rceil \end{pmatrix}, \tag{2.22}$$

where $\delta = 1$ when M is a multiple of 4, and $\delta = 0$ otherwise. In Table 2.1, we use (2.21) and (2.22) to tabulate the maximal rate and minimum delay for complex orthogonal space-time block codes for two to 16 transmit antennas. As can be seen from Table 2.1, not only does the code rate tend to $\frac{1}{2}$ as the number of antennas increases, but the code length becomes prohibitively large for practical implementations. These two drawbacks provided the motivation to construct diagonal algebraic space-time block codes and quasiorthogonal space-time block codes.

2.4.2 Diagonal Algebraic Space-Time Block Codes

Diagonal algebraic space-time (DAST) block codes are a family of linear space-time codes constructed by the use of rotated constellations [27][28]. The word algebraic in the description of DAST codes comes from the fact that rotation matrices used in DAST codes were constructed using algebraic number field theory [56][57]. The word diagonal refers to the structure of the code matrix, wherein the rotated information symbols are spread over the diagonal of the square code matrix. The DAST block codes not only achieve full diversity order for arbitrary number of transmit antennas, but they also achieve a transmission rate of one symbol per channel use. For M transmit antennas, the DAST block codes transmit

Table 2.1: Maximum Rate and Minimum Decoding Delay of Complex Orthogonal Space-Time Block Codes.

Number of Transmit Antennas M	Maximum Rate $\mathcal{R}$	Minimum Delay D_{min}
2	1	2
3	$\frac{3}{4}$	4
4	$\frac{3}{4}$	4
5	$\frac{2}{3}$	15
6	$\frac{2}{3}$	30
7	$\frac{5}{8}$	56
8	$\frac{5}{8}$	56
9	$\frac{3}{5}$	210
10	$\frac{3}{5}$	420
11	$\frac{7}{12}$	792
12	$\frac{7}{12}$	792
13	$\frac{4}{7}$	3003
14	$\frac{4}{7}$	6006
15	$\frac{9}{16}$	11440
16	$\frac{9}{16}$	11440

M information symbols in M signal intervals as follows:

$$\begin{aligned} \mathbf{C}_{DAST}(x_1, \cdots, x_M) &= \text{diag}(\mathbf{u}) \\ &= \begin{bmatrix} u_1 & 0 & \cdots & 0 \\ 0 & u_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & u_M \end{bmatrix}, \end{aligned} \tag{2.23}$$

where

- $\text{diag}(\mathbf{u})$ is the diagonal matrix with the vector $\mathbf{u}$ on the diagonal
- $\mathbf{u} = \mathbf{G}\mathbf{x} = [u_1, u_2, \cdots, u_M]^\top$

- $\mathbf{x} = [x_1, x_2, \cdots, x_M]^\top$ is the vector of complex information symbols drawn from q-ary QAM alphabet
- $\mathbf{G}$ is an $M \times M$ unitary *rotation* or *generator* matrix chosen to ensure a fully diverse DAST code.

The DAST block codes were developed for QAM alphabets and outperform complex orthogonal designs in terms of the SNR required to achieve a target error probability when the number of transmit antennas is greater than two (i.e, $M > 2$). The cost of improved performance compared to complex orthogonal designs is increased decoding complexity.

In general, DAST block codes with QAM alphabet and complex rotation matrices are not separable. Hence, the ML decoder requires detecting all complex symbols jointly. With real rotation matrices, however, the DAST codes are 2-group decodable, wherein the real and imaginary components of the QAM symbols can be decoded separately. Clearly, complex rotation matrices result in higher decoding complexity than real rotations. However, they have a better bit-error-rate performance. For example, for four transmit antennas, complex rotation matrices offer 0.7 dB gain over real rotations [28].

2.4.3 Quasiorthogonal Space-Time Block Codes

Quasiorthogonal space-time codes relax the orthogonality constraint of (2.19) to enable rate-one transmission, at the expense of an increase in decoding complexity. For example, quasiorthogonal codes for four antennas were proposed independently by Jafarkhani [38], Tirkkonen-Boariu-Hottinen [39] and Papadias-Foschini [40]; these codes, however, were not fully diverse.

Full diversity quasiorthogonal space-time block codes were constructed using constellation rotation. For example, rate-one and full-diversity quasiorthogonal codes with rotation were proposed by Tirkkonen [41], Sharma-Papadias [42], and Su-Xia [43]. These quasiorthogonal codes outperformed orthogonal codes at all spectral efficiencies for four transmit antennas in terms of the SNR required to achieve a target error probability. Rate-one and full-diversity quasiorthogonal codes for an arbitrary number of antennas were proposed by Sharma-Papadias [22]. The quasiorthogonal space-time code for M transmit antennas,

where M is a power of two, is constructed from the quasiorthogonal space-time code for $M/2$ antennas as follows [22]

$$\mathbf{C}_{Quasi}(x_1, \cdots, x_M) = \begin{bmatrix} \mathbf{C}_{Quasi}(y_1, \cdots, y_{\frac{M}{2}}) & \mathbf{C}_{Quasi}(y_{\frac{M}{2}+1}, \cdots, y_M) \\ -\mathbf{C}_{Quasi}(y^*_{\frac{M}{2}+1}, \cdots, y^*_M) & \mathbf{C}_{Quasi}(y^*_1, \cdots, y^*_{\frac{M}{2}}) \end{bmatrix}, \tag{2.24}$$

where

- $y_m = e^{i\phi_m} x_m, m \in \{1, \cdots, M\}$, are the rotated q-ary QAM information symbols
- $\mathbf{C}_{Quasi}(y_k, \cdots, y_{k+\frac{M}{2}-1})$ is the quasiorthogonal space-time code for $\frac{M}{2}$ transmit antennas in the rotated symbols y_k through $y_{k+\frac{M}{2}-1}$

By definition, $\mathbf{C}_{Quasi}(x) = x$, and the quasiorthogonal code for two transmit antennas is the orthogonal Alamouti space-time block code [21]. The rotation angles $\phi_m, m \in \{1, \cdots, M\}$, are chosen to ensure full diversity and maximize the coding gain.

For M transmit antennas, where M is not a power of two, the quasiorthogonal space-time code can be obtained by deleting any $\bar{M}-M$ columns of the code matrix $\mathbf{C}_{Quasi}(x_1, \cdots, x_{\bar{M}})$ for $\bar{M}$ antennas, where

$$\bar{M} = 2^{\lceil \log_2 M \rceil}. \tag{2.25}$$

For example, the quasiorthogonal space-time block code for three antennas is obtained by deleting any one column from the code matrix for four antennas. The space-time block codes for three and four transmit antennas are then given by

$$\mathbf{C}_{Quasi} = \begin{bmatrix} x_1 & x_2 & x_3 e^{i\phi} \\ -x_2^* & x_1^* & -x_4^* e^{-i\phi} \\ -x_3^* e^{-i\phi} & -x_4^* e^{-i\phi} & x_1^* \\ x_4 e^{i\phi} & -x_3 e^{i\phi} & -x_2 \end{bmatrix}, \tag{2.26}$$

and

$$\mathbf{C}_{Quasi} = \begin{bmatrix} x_1 & x_2 & x_3 e^{i\phi} & x_4 e^{i\phi} \\ -x_2^* & x_1^* & -x_4^* e^{-i\phi} & x_3^* e^{-i\phi} \\ -x_3^* e^{-i\phi} & -x_4^* e^{-i\phi} & x_1^* & x_2^* \\ x_4 e^{i\phi} & -x_3 e^{i\phi} & -x_2 & x_1 \end{bmatrix}, \tag{2.27}$$

where $\phi = \pi/4$ is the angle that maximizes the coding gain and ensures full diversity. As can be seen from (2.26), the code for three antennas was obtained by deleting the last column of the code matrix (2.27) for four antennas.

Because quasiorthogonal space-time block codes achieve rate-one transmission and full diversity, they outperform orthogonal designs at all spectral efficiencies for complex constellations. Similar to DAST codes, the improved performance comes at the expense of increased decoding complexity compared to orthogonal designs. In particular, quasiorthogonal space-time block codes are 2-group decodable for M antennas. For example, for four transmit antennas, quasiorthogonal designs are 2-group decodable, and each group has two complex symbols. The DAST block codes with real rotations are also 2-group decodable, but each group has four real symbols. By comparison, orthogonal design for four transmit antennas are 6-group decodable, and each group has one real symbol. Clearly, the orthogonal code has the lowest decoding complexity. However, it is not immediately obvious which has the lower complexity, DAST or quasiorthogonal codes. Such situation provides the motivation for presenting a unified framework for discussing worst-case decoding complexity. We defer the discussion to Chapter 3.

2.4.4 Single-Symbol Decodable Space-Time Block Codes

Although orthogonal designs have low decoding complexity, their rate is $\frac{3}{4}$ symbols per signaling interval for three and four transmit antennas. For the particular configuration of three or four transmit antennas, a family of nonorthogonal space-time codes that achieve rate-one transmission and full diversity for three and four transmit antennas was developed. Furthermore, these space-time codes are 4-group decodable, and each group has two real symbols [58]-[61]. Since a pair of real symbols defines a single complex symbol, these codes are said to be *single-symbol decodable*. Single-symbol decodable codes have higher complexity than orthogonal designs, but offer higher rate. In comparison to quasiorthogonal and DAST codes, they offer the same rate, but at a lower decoding complexity.

A framework for the construction of single-symbol decodable space-time codes was presented in [61]. The encoder decomposes into a concatenation of three steps, as shown in

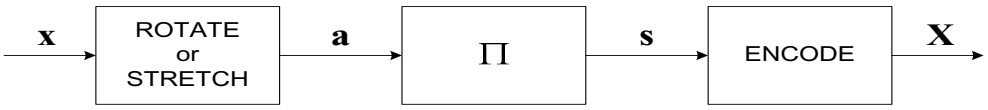


Figure 2.2: Encoding architecture of single-symbol decodable space-time block codes.

Figure 2.2. The encoder starts with a vector of information symbols $\mathbf{x} = [x_1, x_2, x_3, x_4]^\top$ chosen from a conventional q-ary QAM alphabet. The first step is to distort the alphabet in some way; the codes of [58] and [59] rotate each alphabet by an angle of ϕ such that $a_k = x_k e^{i\phi}$, $k \in \{1, \cdots, 4\}$ while the code of [61] stretches the alphabet by a factor $\mathcal{K}$ such that $a_k = \sqrt{\frac{2}{1+\mathcal{K}^2}}(\mathcal{K}x_k^R + ix_k^I)$. The rotation angle and scaling factor are chosen to maximize the coding gain. The purpose of the rotation or scaling is to ensure full diversity. The second step is to interleave the coordinates of $\mathbf{a}$, yielding $\mathbf{s} = \Pi(\mathbf{a})$. The interleavers act on the real and imaginary parts separately, so that $[s_1^R, s_1^I, s_2^R, s_2^I, s_3^R, s_3^I, s_4^R, s_4^I] = [a_1^R, a_1^I, a_2^R, a_2^I, a_3^R, a_3^I, a_4^R, a_4^I]\Pi$, where Π is an 8×8 permutation matrix (so that its columns are a permutation of the columns of the identity matrix, with the possibility of sign inversion), and where s_k^R and s_k^I denote the real and imaginary parts, respectively, of s_k. The interleaver is used to achieve full diversity while maintaining single-symbol decodability for the codes of [58] and [59] and also to ensure that the peak-to-average power ratio is the same as the underlying QAM alphabet for the code of [61]. The final step is to encode $\mathbf{s}$ using a conventional space-time block encoder $\mathbf{G}(\cdot)$, yielding $\mathbf{C} = \mathbf{G}(\mathbf{s})$. We next give an example of the single-symbol decodable code of [59] for four antennas.

In terms of Figure 2.2, the code of [59] is specified by $\phi = \tan^{-1}(2)$, $\Pi = [\mathbf{e}_1, \mathbf{e}_6, \mathbf{e}_3, \mathbf{e}_8, \mathbf{e}_5, \mathbf{e}_2, \mathbf{e}_7, \mathbf{e}_4]$, where $\mathbf{e}_i$ is the i-th column of the 8×8 identity matrix, and

$$\mathbf{G}(\mathbf{s}) = \sqrt{2}\begin{bmatrix} \mathbf{A}(s_1, s_2) & \mathbf{0} \\ \mathbf{0} & \mathbf{A}(s_3, s_4) \end{bmatrix} \tag{2.28}$$

where $\mathbf{A}(s_1, s_2) = \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix}$ is the Alamouti space-time code [21]. The constant $\sqrt{2}$

ensures that the average transmit power is identical to that of the underlying alphabet $\mathcal{A}$. Therefore, in terms of the rotated information symbols $a_k = e^{i\phi}x_k$, the single-symbol decodable space-time code of [59] is

$$\mathbf{C}_{SSD} = \sqrt{2}\begin{bmatrix} a_1^R + ia_3^I & a_2^R + ia_4^I & 0 & 0 \\ -a_2^R + ia_4^I & a_1^R - ia_3^I & 0 & 0 \\ 0 & 0 & a_3^R + ia_1^I & a_4^R + ia_2^I \\ 0 & 0 & -a_4^R + ia_2^I & a_3^R - ia_1^I \end{bmatrix}. \tag{2.29}$$

2.4.5 Semi-Orthogonal Space-Time Block Codes

We next discuss the semi-orthogonal algebraic space-time (SAST) block codes, a family of linear space-time codes that achieve rate-one transmission and full diversity for any number of transmit antennas [23]. The word algebraic in the description of SAST codes comes from the fact that they use the same real rotation matrices of the DAST codes [56][57] (see also [62]), which were constructed using algebraic number field theory. In fact, SAST codes are constructed using the DAST codes, as we will discuss shortly. The word semi-orthogonal refers to a property of the code matrix, where half the columns of the code matrix are orthogonal to the other half.

An equivalent form of the semi-orthogonal algebraic space-time block code for M transmit antennas, where M is even, is [23]

$$\mathbf{C}_{SAST} = \begin{bmatrix} \mathbf{C}_{DAST}(x_1, \cdots, x_{\frac{M}{2}}) & \mathbf{C}_{DAST}(x_{\frac{M}{2}+1}, \cdots, x_M) \\ -\mathbf{C}_{DAST}(x^*_{\frac{M}{2}+1}, \cdots, x^*_M) & \mathbf{C}_{DAST}(x^*_1, \cdots, x^*_{\frac{M}{2}}) \end{bmatrix}. \tag{2.30}$$

The SAST code for $M-1$ antennas is obtained by deleting the last column of the SAST code for M antennas. This version of SAST codes differs from the original in [23] only by the fact that we used the DAST matrices instead of using circulant matrices. The circulant matrix reduces the peak-to-average-power ratio but has no impact on the diversity, coding gain, and decoding complexity. In fact, any circulant matrix can be diagonalized using the Fourier transform matrix. We simply use DAST matrices in (2.30) to simplify the presentation.

The semi-orthogonal property of SAST codes allows for the separate decoding of symbols x_1 through $x_{\frac{M}{2}}$ from symbols $x_{\frac{M}{2}+1}$ through x_M. Furthermore, the use of real rotation matrices allows for separate decoding of the real and imaginary parts of the QAM symbols. As a result, the SAST block codes for M and $M-1$ antennas require the joint decoding of only $\frac{M}{2}$ real PAM symbols. For the case of two transmit antennas, the SAST code simplifies to the Alamouti space-time block code. For the case of three and four transmit antennas, the SAST codes require the joint decoding of only two real symbols, which is the same decoding complexity as the single-symbol decodable codes. In fact, the SAST code for four transmit antenna is equivalent to the single-symbol decodable code of [58] and has identical diversity order, coding gain and bit-error-rate performance. The SAST codes have the lowest decoding complexity of any rate-one space-time block code and achieve comparable bit-error-rate performance to the best codes.

2.4.6 Threaded Algebraic Space-Time Block Codes

All of the space-time codes that we have discussed thus far achieve a maximum rate of one symbol per signaling interval. We next discuss the first of two families of high-rate space-time block codes: the threaded algebraic space-time (TAST) block codes. The threaded algebraic space-time codes are a family of linear space-time codes that are fully diverse and achieve arbitrary rate for arbitrary number of transmit antennas. The TAST block codes *layer* or *thread* rate-one DAST block codes to achieve maximal rate of M symbols per signaling interval for M transmit antennas. As a result, they can also achieve other rates by simply puncturing or deleting layers. For example, for a four transmit and four receive antenna system, the TAST block codes of rate $\mathcal{R} \in \{1, 2, 3, 4\}$ are easily obtained by puncturing or deleting $4 - \mathcal{R}$ layers.

We next discuss the construction of TAST block codes. We begin by presenting the notation that will be used in describing the TAST codes, the perfect space-time block codes and the proposed embedded-orthogonal space-time block codes.

The TAST code encodes $K = \mathcal{R}M$ complex information symbols $\{x_{\ell,m}\}$, drawn from a q-ary QAM alphabet, that are organized into $\mathcal{R}$ threads, where $\mathbf{x}_\ell = [x_{\ell,1}, x_{\ell,2}, \cdots, x_{\ell,M}]^\top$

denotes the information symbols for the ℓ-th thread, $\ell \in \{1, \cdots, \mathcal{R}\}$. The rate-$\mathcal{R}$ TAST code can be written as [4]

$$\mathbf{C}_{TAST} = \sum_{\ell=1}^{\mathcal{R}} \text{diag}(\mathbf{u}_\ell)\mathbf{J}^{\ell-1}, \tag{2.31}$$

where

- $\text{diag}(\mathbf{u}_\ell)$ is the diagonal matrix with the vector $\mathbf{u}_\ell$ on the diagonal
- $\mathbf{u}_\ell = \mathbf{G}\mathbf{x}_\ell = [u_{\ell,1}, u_{\ell,2}, \cdots, u_{\ell,M}]^\top$
- $\mathbf{G}$ is an $M \times M$ unitary *rotation* or *generator* matrix
- $\mathbf{J} = [\phi\mathbf{e}_M, \mathbf{e}_1, \cdots, \mathbf{e}_{M-1}]$
- $\mathbf{e}_m$ is the m-th column of the $M \times M$ identity matrix
- ϕ is a unit-magnitude complex number, the value of which depends on both the QAM modulation alphabet size and number of transmit antennas M.

We make two important remarks regarding the construction of TAST codes. First, the TAST code simplifies to the DAST code for $\mathcal{R} = 1$. This result is expected since the TAST code threads or layers the rate-one DAST code to obtain the higher transmission rates. Second, the design of a TAST code is a two-step design process. The first step is to choose an algebraic rotation matrix $\mathbf{G}$. The second step is to choose ϕ to ensure full diversity. The two-step design process simplifies the design problem, since the algebraic rotation matrices $\mathbf{G}$ that maximize the coding gain of an encoded thread have been thoroughly investigated in the literature in the context of DAST codes.

The decoding complexity of TAST codes depends on the transmission rate and whether complex or real generator matrices are used. When complex rotation matrices are used, the TAST code does not offer any reduction in decoding complexity. This result is not surprising since TAST codes layer DAST codes, and DAST codes with complex rotation matrices do not offer any reduction in decoding complexity relative to an exhaustive-search ML decoder. With real generator matrices, however, a reduction in decoding complexity is possible. We will defer the discussion of worst-case ML decoding complexity to Chapter 3.

This is because for $\mathcal{R} > 1$, the TAST codes are not separable and the worst-case decoding complexity for an exhaustive-search ML decoder is $q^{\mathcal{R}M}$. In Chapter 3, however, we will show that TAST codes with real generator matrices allow for a reduction in the worst-case ML decoding complexity relative to the exhaustive-search ML decoder.

2.4.7 Perfect Space-Time Block Codes

The second family of high-rate space-time block codes is the perfect codes. Perfect space-time block codes are a family of linear space-time codes that were proposed for two, three, four, and six antennas in [19] and later generalized for any number of antennas in [20]. These codes are termed perfect because they have full diversity, a nonvanishing determinant for increasing spectral efficiency, uniform average transmitted energy per antenna and achieve rate M symbols per signaling interval for M antennas. Similar to TAST codes, perfect space-time codes layer or thread rate-one diagonal codes to achieve a maximal rate of M symbols per signaling interval and can also achieve other rates by puncturing layers. We next discuss encoding of the rate $\mathcal{R}$ perfect code.

The rate-$\mathcal{R}$ perfect code is also given by (2.31). We reproduce the results here for completeness.

$$\mathbf{C}_{Perfect} = \sum_{\ell=1}^{\mathcal{R}} \mathrm{diag}(\mathbf{u}_\ell)\mathbf{J}^{\ell-1}, \tag{2.32}$$

where

- $\mathrm{diag}(\mathbf{u}_\ell)$ is the diagonal matrix with the vector $\mathbf{u}_\ell$ on the diagonal
- $\mathbf{u}_\ell = \mathbf{G}\mathbf{x}_\ell = [u_{\ell,1}, u_{\ell,2}, \cdots, u_{\ell,M}]^\top$
- $\mathbf{G}$ is an $M \times M$ unitary *rotation* or *generator* matrix
- $\mathbf{J} = [\phi\mathbf{e}_M, \mathbf{e}_1, \cdots, \mathbf{e}_{M-1}]$
- $\mathbf{e}_m$ is the m-th column of the $M \times M$ identity matrix
- ϕ is a unit-magnitude complex number, the value of which depends on the modulation alphabet and number of transmit antennas M.

- The alphabet is HEX for the case of three and six transmit antennas. For other configurations, the alphabet is QAM.

Unlike the TAST codes, the parameter ϕ for the perfect codes does not depend on the modulation alphabet size. This is expected since the perfect codes satisfy the nonvanishing determinant property. The TAST and perfect codes differ in the choice of the three parameters; $\mathbf{G}$, ϕ and the modulation alphabet, although some choices are the same.

The decoding complexity of perfect space-time codes not only depends on the transmission rate, but also on the modulation alphabet. Similar to TAST codes, the perfect codes are not separable for $\mathcal{R} > 1$, and the worst-case ML decoding complexity for an exhaustive-search decoder is $q^{\mathcal{R}M}$. In Chapter 3, we will also show that perfect codes allow for a reduction in the worst-case ML decoding complexity for q-ary QAM alphabet.

2.5 Conclusions

In this chapter, we reviewed the design criteria of space-time codes, and we discussed the construction of some important families of space-time block codes. In general, the design of space-time block codes is a tradeoff between three conflicting goals: minimizing the decoding complexity, maximizing the diversity gain to improve reliability and maximizing the multiplexing gain to improve throughput. On one extreme, we have the orthogonal designs, which minimize the decoding complexity and maximize the diversity gain at the expense of the multiplexing gain. On the other extreme, we have the perfect space-time block codes, which achieve the optimal diversity-multiplexing tradeoff at the expense of decoding complexity, which is exponential in the number of information symbols for an exhaustive-search ML decoder.

CHAPTER 3

A UNIFIED FRAMEWORK FOR DETERMINING WORST-CASE MAXIMUM-LIKELIHOOD DECODING COMPLEXITY

An important design goal for a space-time code is that its worst-case ML decoding complexity is less than q^K. The design of space-time block codes with reduced decoding complexity and good error rate performance is one of the main contribution of this research. In this chapter, we present a unified framework for discussing the worst-case ML decoding complexity of space-time block codes.

The framework for discussing the worst-case ML decoding complexity is based on the observation that the structure of the space-time block code induces certain properties in the effective channel matrix. As a result, we will examine the worst-case ML decoding complexity in terms of the properties of the effective channel matrix. Specifically, we will examine the properties of the $\mathbf{R}$ matrix in the orthogonal-triangular (QR) decomposition of the effective channel matrix. A reduction in the worst-case ML decoding complexity is made possible by exploiting certain properties of the $\mathbf{R}$ matrix.

The remainder of the chapter is organized as follows. In Section 3.1, we review the orthogonal-triangular (QR) decomposition of the effective channel matrix. In Section 3.2, we review the sphere decoding algorithm, which is a tree search decoding algorithm. In Section 3.3, we discuss the key properties for reduced complexity decoding. In Section 3.4, we present the general form of the $\mathbf{R}$ matrix for the different families of space time block codes including orthogonal, quasiorthogonal, diagonal algebraic, single-symbol decodable, semi-orthogonal algebraic, threaded algebraic and perfect space-time block codes. In Section 3.5, we compare the different families of space-time block codes in terms of their worst-case ML decoding complexity, as well as their code rate and minimum delay. In Section 3.6, we conclude the chapter.

3.1 The QR Decomposition of the Effective Channel Matrix

A QR decomposition of the matrix $\mathbf{H}$ can be obtained by applying the Gram-Schmidt procedure to the columns of $\mathbf{H} = [\mathbf{h}_1, \cdots, \mathbf{h}_K]$ to obtain $\mathbf{H} = \mathbf{QR}$, where the columns of $\mathbf{Q} = [\mathbf{q}_1, \cdots, \mathbf{q}_K]$ are an orthonormal basis for the subspace spanned by $\mathbf{H}$, and $\mathbf{R}$ is upper triangular with nonnegative real diagonal elements, so that the entry of $\mathbf{R}$ in row i and column j is $r_{i,j} = \mathbf{q}_i^* \mathbf{h}_j$. The classical Gram-Schmidt procedure is shown in figure 3.1.

Algorithm: `Classical Gram-Schmidt`

```
1 for k from 1 to K do
2 |   v_k = h_k
3 |   for l from 1 to k − 1 do
4 |   |   r_{l,k} = q_l^* h_k
5 |   |   v_k = v_k − r_{l,k} q_l
6 |   end
7 |   r_{k,k} = ‖v_k‖
8 |   q_k = v_k / r_{k,k}
9 end
```

Figure 3.1: Classical Gram-Schmidt.

The classical Gram-Schmidt procedure is conceptually simple, but it suffers from two drawbacks. First, it is numerically unstable due to rounding errors using finite-precision arithmetic [63]. The second drawback is that it does not explicitly deal with the case of a rank deficient $\mathbf{H}$ matrix, which occurs when there is at least one column that can be expressed as a linear combination of other columns. Linear dependency leads to a situation where $\mathbf{v}_k$ (in line 5) is the zero vector. Consequently, $r_{k,k}$ in line 7 is zero, and we have a divide by zero operation on line 8.

Both of these drawbacks have been addressed extensively in the mathematical literature. An excellent discussion of the different QR decomposition algorithms can be found in [64]. Algorithms based on modifying the classical Gram-Schmidt procedure solve the numerical stability and rank deficiency constraint of the Gram-Schmidt algorithm. The interested reader is referred to [64] for pseudocode and MATLAB® source code for the different QR decomposition algorithms including the modified Gram-Schmidt, the modified Gram-Schmidt with reorthogonalization and Householder triangularization.

Because $\mathbf{Q}$ satisfies $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, the ML decoder minimizes the cost function

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \\ &= \|\mathbf{y} - \mathbf{Q}\mathbf{R}\mathbf{x}\|^2 \\ &= \|\mathbf{Q}^*\mathbf{y} - \mathbf{R}\mathbf{x}\|^2 \\ &= \|\mathbf{z} - \mathbf{R}\mathbf{x}\|^2, \end{aligned} \tag{3.1}$$

where

$$\mathbf{z} = \mathbf{Q}^*\mathbf{y}. \tag{3.2}$$

Because the ML cost function depends on $\mathbf{R}$, the worst-case decoding complexity also depends on the properties of the $\mathbf{R}$ matrix.

For the real-valued system model in (2.7), we also examine the properties of the $\mathbf{R}$ matrix in the orthogonal-triangular decomposition of the matrix $\check{\mathbf{H}} = [\check{\mathbf{h}}_1, \cdots, \check{\mathbf{h}}_{2K}]$. We again have $\check{\mathbf{H}} = \mathbf{Q}\mathbf{R}$, where the columns of $\mathbf{Q} = [\mathbf{q}_1, \cdots, \mathbf{q}_{2K}]$ are an orthonormal basis for the subspace spanned by $\check{\mathbf{H}}$, and $\mathbf{R}$ is upper triangular with nonnegative real diagonal elements, so that the entry of $\mathbf{R}$ in row i and column j is $r_{i,j} = \mathbf{q}_i^\top \check{\mathbf{h}}_j$. Finally, the ML decoder minimizes the cost function

$$\begin{aligned} P(\check{\mathbf{x}}) &= \|\check{\mathbf{y}} - \check{\mathbf{H}}\hat{\mathbf{x}}\|^2 \\ &= \|\check{\mathbf{y}} - \mathbf{Q}\mathbf{R}\check{\mathbf{x}}\|^2 \\ &= \|\mathbf{Q}^\top\check{\mathbf{y}} - \mathbf{R}\check{\mathbf{x}}\|^2 \\ &= \|\check{\mathbf{z}} - \mathbf{R}\check{\mathbf{x}}\|^2, \end{aligned} \tag{3.3}$$

where

$$\check{\mathbf{z}} = \mathbf{Q}^\top\check{\mathbf{y}}. \tag{3.4}$$

3.2 The Sphere Decoder

In Chapter 2, we defined the worst-case ML decoding complexity of an exhaustive-search decoder for space-time block codes. In this section, we will introduce another measure of

decoding complexity, which is the *average number of nodes* visited in a tree-based decoding algorithm. We next discuss a tree-based search algorithm, namely the sphere decoder.

Let us re-examine the cost function in (3.1), which we write as:

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{z}-\mathbf{R}\mathbf{x}\|^2 \\ &= \sum_{k=1}^{K}\left|z_k-\sum_{l=k}^{K} r_{k,l}x_l\right|^2 \\ &= \underbrace{\left|z_1-\sum_{l=1}^{K} r_{1,l}x_l\right|^2}_{P_1}+\underbrace{\left|z_2-\sum_{l=2}^{K} r_{2,l}x_l\right|^2}_{P_2}+\cdots+\underbrace{|z_K-r_{K,K}x_K|^2}_{P_K}. \end{aligned} \tag{3.5}$$

The cost function in (3.5) is the sum of K *branch metrics*, where the k-th branch metric, $k \in \{1,\cdots,K\}$, is given by

$$P_k = \left|z_k-\sum_{l=k}^{K} r_{k,l}x_l\right|^2. \tag{3.6}$$

With this interpretation, the cost function in (3.5) is then referred to as the *path metric*. The goal of the ML decoder is to find $\mathbf{x}$ with the smallest path metric. Let us associate the path metric with a *search tree*, where there are a total of q^K paths to traverse the tree, which correspond to the q^K possible choices for $\mathbf{x}$. We next discuss the search tree and define important terms related to tree search. We begin the discussion with an example.

Example 3.1. Consider a space-time block code transmitting three complex information symbols from 4-QAM alphabet. The cost function or path metric of the space-time block code is given by:

$$\begin{aligned} P(\mathbf{x}) &= P(x_1,x_2,x_3) \\ &= \underbrace{|z_1-(r_{1,1}x_1+r_{1,2}x_2+r_{1,3}x_3)|^2}_{P_1}+\underbrace{|z_2-(r_{2,2}x_2+r_{2,3}x_3)|^2}_{P_2}+\underbrace{|z_3-r_{3,3}x_3|^2}_{P_3}. \end{aligned} \tag{3.7}$$

Note the path metric was written as the sum of three branch metrics. This space-time block code, however, is not separable since these branch metrics are not independent of each other. For example, P_1 depends on x_1, but so does P_2 and P_3. The tree search is shown in Figure 3.2. We first discuss a few terms related to the tree and then discuss how tree search is performed.

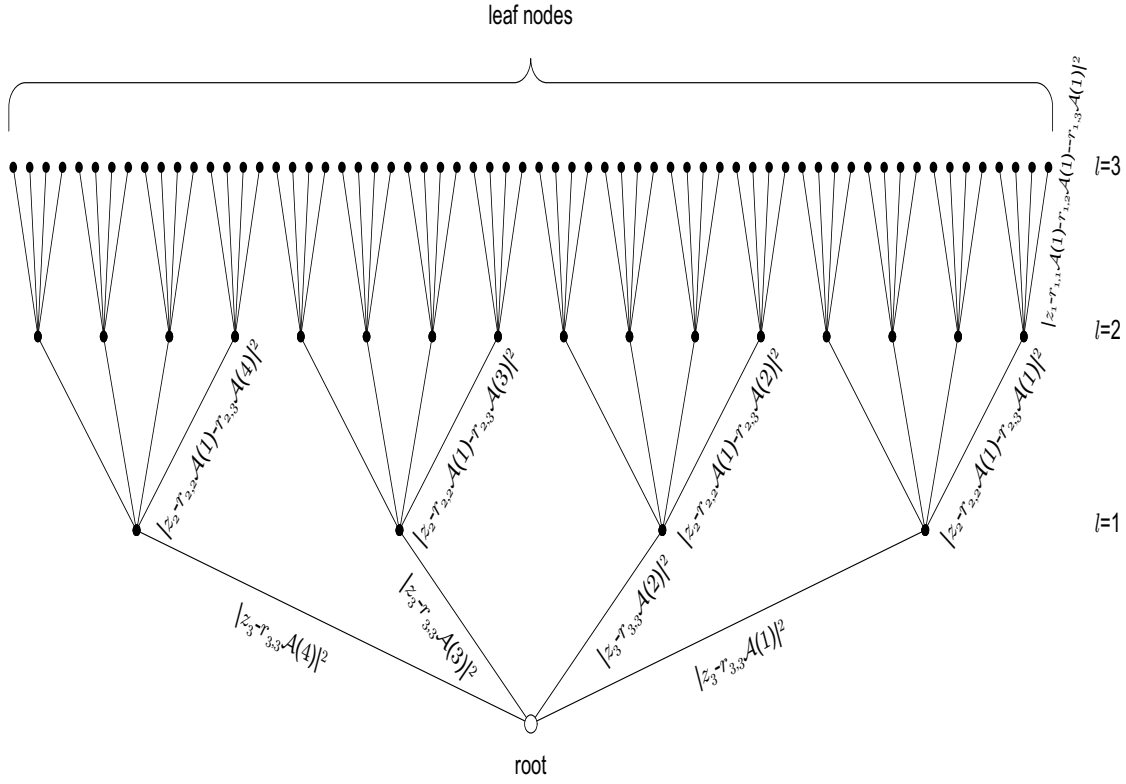


Figure 3.2: Three level tree search with $q = 4$.

The base of the tree is known as the *root*. There are $4^1 + 4^2 + 4^3 = 84$ branches in the tree. The end point of each branch is known as a *node*. Therefore, there are also 84 nodes in the tree. The tree is organized into three *levels* or *layers*. The number of levels in the tree corresponds to the number of symbols detected in the tree search. The first, second and third levels of the tree have 4, 16 and 64 branches, respectively. In general, for the l-th level in the tree, there are q^l branches. The nodes at the l-th layer in the tree stemming from the node at the $(l-1)$-th layer, termed *parent node*, are known as *child nodes*. For example, there are 4 child nodes stemming from the root of the tree, and everyone of those child nodes is a parent node, which has 4 child nodes of its own. The nodes at the last level of the tree, which do not have any child nodes, are known as *leaf nodes*. Child nodes are also *sibling nodes* to each other. We next discuss how the tree search is performed.

The tree search begins at the root of the tree. We ascend the tree by selecting the node with the smallest branch metric, out of the four possible child nodes. We continue ascending the tree until we reach a leaf node. The path from the root of the tree all the way to a leaf node corresponds to the first candidate solution. Both the path metric along with the candidate solution are stored as potentially the smallest path metric and ML solution.

The tree search continues by descending to the second level and considering the remaining three sibling nodes, which are siblings to the leaf node and children of the node at the second level in the tree. A natural question to ask is how to select one node out of the remaining three sibling nodes. We will use *Schnorr-Euchner enumeration*, which chooses the siblings in ascending order of their branch metric weight. We will discuss later in the section how Schnorr-Euchner enumeration, also known as *sorting*, allows us to visit fewer leaf nodes compared to a strategy based on picking the siblings based on their natural position in the alphabet. For the sake of this example, we will simply treat sorting as a strategy for picking among sibling nodes.

After selecting a sibling node, we determine the corresponding path metric, and compare to the stored path metric. If the new path metric is smaller than the stored path metric, then the new path metric and new solution replace the stored path metric and current candidate ML solution. If the new path metric is greater than the stored path metric, it is

simply discarded. We continue this process until we consider all sibling nodes. Then, we descend one more level down the tree so that we are now at the first level. We ascend up the tree by considering one of the three siblings at the first level. We continue ascending up and descending down the tree until all leaf nodes have been considered. The ML solution corresponds to the path from the root to a leaf node with the smallest path metric. □

The complexity of the tree search decoder, as discussed in Example 3.1 is the same as that of an exhaustive-search decoder. Specifically, in finding the ML solution, we computed $q^K = 4^3 = 64$ path metrics, which is the same as that of an exhaustive-search decoder. The complexity of a tree search can be significantly reduced by incorporating tree *pruning*, a strategy that allows us to discard branches in the tree. We next discuss an example that shows how sorting, in conjunction with pruning can reduce the decoding complexity.

Example 3.2. Consider the tree search problem in Figure 3.3. The number on each branch represents its branch metric. Starting at the root of the tree, we choose the branch metric that has weight 1 at level 1. Then we choose the branch metric that has weight 5 at level 2. Finally, we reach the leaf node by selecting the branch metric that has weight 4. We store the leaf node $[1, 5, 4]$ along with the path metric $1 + 5 + 4 = 10$. Because of sorting, the branch metric of each of the remaining three siblings is $4 + \delta$, for some $\delta > 0$. Therefore, we do not need to consider any siblings of the leaf node, since the cost of their leaf nodes is $1 + 5 + 4 + \delta > 10$. We descend the tree to level 1 and consider the branch metric with weight 6. However, the leaf node with smallest branch metric is 5, so the path metric is $1 + 6 + 5 = 11 > 10$. Therefore, we can discard the leaf node along with its siblings. We descend again to level 1. The cost of the next sibling is $17 > 10$, so we can discard this node, along with all its children. Similarly, we discard the last sibling, with branch metric weight $18 > 10$ along with all its children.

We next descend to the root of the tree and consider the next node with branch metric 2. We ascend the tree to level 1 and choose the node with branch metric 1. We ascend the tree to level 2 and choose the leaf node with branch metric 2. The total path metric is $1 + 2 + 2 = 5 < 10$. Therefore, we store the new leaf node $[1, 2, 2]$ along with its path metric

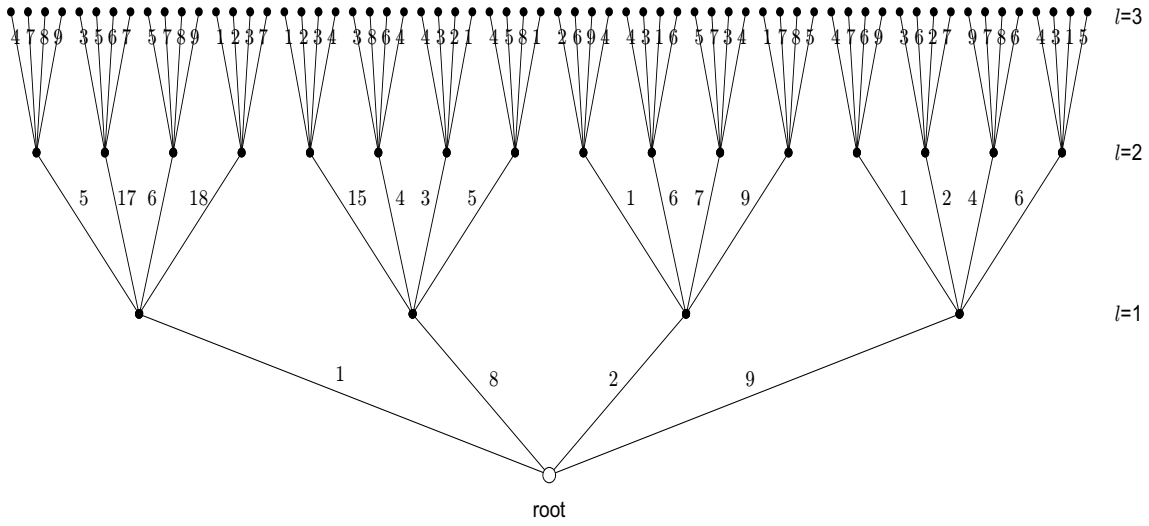


Figure 3.3: Example of three level tree search with $q = 4$ and computed branch metrics.

5. Sorting allows us to discard the siblings of this leaf node and descend to level 1. The next sibling at level 1 is the node with branch metric $6 > 5$. Hence, we can discard this sibling along with all its children. Furthermore, we can discard the remaining siblings since their cost is $6 + \delta > 5$. We descend one more level to the root of tree. The next child node at the root of the tree is the node with branch metric $8 > 5$. Therefore, we can discard this node and the remaining sibling. Our ML solution is the one that corresponds to leaf node [1,2,2] and its cost function or path metric is 5. □

We discussed in Example 3.2 how sorting allows us to prune or discard branches in the tree search since any path that goes through those discarded branches will have a weight that is greater than the current smallest path metric. Therefore, none of those discarded branches can lead to the ML solution.

Tree search decoding that incorporates pruning is referred to as *sphere decoding*. Specifically, tree pruning in the sphere decoding algorithm limits the candidate symbols to those that obey the following sphere constraint

$$\begin{aligned} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 &= \|\mathbf{z} - \mathbf{R}\mathbf{x}\|^2 \\ &< \hat{P}, \end{aligned} \tag{3.8}$$

where $\hat{P}$ is known as the *sphere radius*. The sphere constraint in (3.8) limits the decoder to

visiting only those points for which $\mathbf{Hx}$ lies within a radius $\hat{P}$ of the received vector $\mathbf{y}$. In our examples, we implicitly assumed that the sphere radius is initialized to ∞. Furthermore, we updated the sphere radius with the smallest path metric value encountered so far to prune the tree even more efficiently. If the sphere radius is initialized to a value that is too small, then we might not reach any leaf node because we prune all the branches before reaching a leaf node. Therefore, to guarantee that we reach the ML solution, the sphere radius is initialized to ∞. A pseudocode that implements the sphere decoding for the three level tree in Examples 3.1 and 3.2 is shown in Figure 3.4. We next discuss the pseudocode in detail.

```
   Algorithm: [P̂, x̂] = Sphere Decoder(y, H)
 1 [Q, R] = QR decomposition(H)
 2 z = Q*y
 3 P̂ = ∞
 4 [P_3, Π_3] = sort_{a∈A}(|z_3 − r_{3,3}a|^2)
 5 for n_3 from 1 to q do
 6 |   if P_3(n_3) > P̂ then
 7 |   |   break
 8 |   end
 9 |   x_3 = A(Π_3(n_3))
10 |   [P_2, Π_2] = sort_{a∈A}(|z_2 − r_{2,3}x_3 − r_{2,2}a|^2)
11 |   for n_2 from 1 to q do
12 |   |   if (P_2(n_2) + P_3(n_3)) > P̂ then
13 |   |   |   break
14 |   |   end
15 |   |   x_2 = A(Π_2(n_2))
16 |   |   x_1^R = Q(ℜ(z_1 − r_{1,2}x_2 − r_{1,3}x_3)/r_{1,1})
17 |   |   x_1^I = Q(ℑ(z_1 − r_{1,2}x_2 − r_{1,3}x_3)/r_{1,1})
18 |   |   P_1 = |z_1 − r_{1,1}x_1 − r_{1,2}x_2 − r_{1,3}x_3|^2
19 |   |   P = P_1 + P_2(n_2) + P_3(n_3)
20 |   |   if P < P̂ then
21 |   |   |   P̂ = P
22 |   |   |   x̂ = [x_1, x_2, x_3]
23 |   |   end
24 |   end
25 end
```

Figure 3.4: Sphere decoder with $K = 3$ tree levels.

The first three lines represent initializations. Specifically, the first step is to perform a QR decomposition on the matrix $\mathbf{H}$ to obtain $\mathbf{H} = \mathbf{QR}$ (line 1). Then, we form the variable

$\mathbf{z}$ in (3.2) in line 2 and initialize the sphere radius to ∞ in line 3. We next begin the tree search.

We start with sorting at the root of the tree (line 4). The function `sort`($\mathbf{a}$) returns the weights of the sorted list in ascending order of distance to the input argument $\mathbf{a}$, along with an index list Π such that $\mathbf{a}_{\Pi(k)}$ is the k-th element in the sorted list. For example, `sort`($[60, 50, 80, 70]$) returns the ordered list $[50, 60, 70, 80]$ and index list $[2, 1, 4, 3]$. Tree pruning is implemented in line 7 and also line 13, where we prune the node and its children if the sum of the branch metrics up to the node is greater than the sphere radius. We pick the best child nodes of the root node in ascending order of their branch metrics in line 9. For every such choice of the parent node at level 1, we perform another sort operation to obtain an ordered list of its child nodes at level 2 of the tree. We choose the child nodes in ascending order of their branch metrics in line 15. For a given choice of the parent node at level 1, its child node at level 2, we determine the leaf node at level 3 with two PAM slicers (lines 16 and 17). We will discuss the implementation of a PAM slicer later in the section. The branch metric for the leaf node and its corresponding path metric are computed in line 18 and line 19, respectively. If the current path metric is less than the sphere radius (line 20), then sphere radius and best candidate symbol vector are updated in line 21 and line 22, respectively.

Having discussed tree search decoding in general and the sphere decoding algorithm in particular, we now define another measure of decoding complexity commonly used in tree search algorithms, the *average node count* or *average complexity*.

Definition 3.1. *Average complexity* in a tree search decoder refers to the average number of nodes visited in the tree search to reach the ML decision.

In practical systems which are designed for the worst-case scenario, the worst-case ML decoding complexity is a more relevant measure than average complexity. However, in some situations, the system cannot be designed for the worst-case due to the fact that the worst-case complexity is computationally prohibitive for practical implementation. In this case, it is also desirable to not only consider worst-case complexity, but also average

complexity. This is because under certain complexity constraints, an algorithm with lower average complexity can be expected to outperform a competing algorithm with higher average complexity. For example, we can impose a run-time constraint on the complexity of the sphere decoding algorithm by terminating the tree search when the number of nodes visited reaches or exceeds a certain threshold. In this case, we can expect that an algorithm that visits fewer nodes, on average, will outperform an algorithm that visits more nodes under the same run-time constraint. Therefore, we will not only consider worst-case decoding complexity, but we will also consider average complexity in our simulation results in Chapters 4, 5 and 6.

We next discuss the properties of the $\mathbf{R}$ matrix in the QR decomposition of the effective channel matrix $\mathbf{H}$ or $\check{\mathbf{H}}$ that lead to a reduction in the worst-case decoding complexity.

3.3 Key Properties for Reduced-Complexity Decoding

We discuss the key properties of reduced complexity decoding of space-time block codes in terms of the resulting $\mathbf{R}$ matrix of the QR decomposition of the complex-valued effective channel matrix $\mathbf{H}$ or the real-valued effective channel matrix $\check{\mathbf{H}}$. We start the discussion with an example.

Example 3.3. Consider the QR decomposition of the effective channel matrix for the Alamouti space-time block code discussed in Example 2.2. The complex-valued effective channel matrix is given by

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{2,1} \\ h_{2,1}^* & -h_{1,1}^* \end{bmatrix}. \tag{3.9}$$

The Alamouti space-time block code induces orthogonality in its effective channel matrix such that the first column is orthogonal to the second column. Specifically, $\mathbf{h}_1^*\mathbf{h}_2 = h_{1,1}^* h_{2,1} - h_{2,1}h_{1,1}^* = 0$. It then follows that the QR decomposition of $\mathbf{H}$ yields

$$\mathbf{H} = \underbrace{\frac{1}{\sqrt{|h_{1,1}|^2 + |h_{2,1}|^2}} \begin{bmatrix} h_{1,1} & h_{2,1} \\ h_{2,1}^* & -h_{1,1}^* \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{bmatrix} \sqrt{|h_{1,1}|^2 + |h_{2,1}|^2} & 0 \\ 0 & \sqrt{|h_{1,1}|^2 + |h_{2,1}|^2} \end{bmatrix}}_{\mathbf{R}}. \tag{3.10}$$

Notice that each column in $\mathbf{Q}$ is a normalized version of the columns of $\mathbf{H}$. This is expected since the columns of $\mathbf{H}$ are orthogonal.

The decoding complexity of the Alamouti space-time block code is easily seen from the $\mathbf{R}$ matrix. Because the element $r_{1,2}$ is zero, there is no interference between the symbols x_1 and x_2, and hence, we can decode them separately. Furthermore, because $r_{1,1}$ and $r_{2,2}$ are real, the real and imaginary components of x_1 and x_2 are also separately decodable, assuming a QAM alphabet. More formally, the ML decision minimizes

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{z} - \mathbf{R}\mathbf{x}\|^2 \\ &= |z_1 - r_{1,1}x_1|^2 + |z_2 - r_{2,2}x_2|^2 \\ &= |z_1^R - r_{1,1}x_1^R|^2 + |z_1^I - r_{1,1}x_1^I|^2 + |z_2^R - r_{2,2}x_2^R|^2 + |z_2^I - r_{2,2}x_2^I|^2, \end{aligned} \tag{3.11}$$

where the second equality follows from the fact that $r_{1,2}$ is zero, and the third equality follows from the fact that $r_{1,1}$ and $r_{2,2}$ are real. Hence, all the transmitted symbols are decoded separately. □

As discussed in Example 3.3, the decoding complexity of the Alamouti space-time block code is the same as the decoding complexity of a PAM symbol. We next discuss the decoding complexity of a single transmitted symbol from a $\sqrt{q}$-ary PAM or q-ary QAM alphabet.

The decoding complexity of a QAM or PAM symbol does not grow as the size of the alphabet grows, and hence, the decoding complexity is $\mathcal{O}(1)$. This is because we can decode a QAM symbol using a pair of PAM slicers, and the complexity of each slicer is independent of the size of the alphabet q. Specifically a PAM slicer $Q(x)$, where x is a real symbol drawn from the $\sqrt{q}$-ary PAM alphabet $\{\pm 1, \pm 3, \cdots, \pm(\sqrt{q}-1)\}$ is given by:

$$Q(x) = \min\left\{\max\left\{2 \times \texttt{round}\left(\frac{x-1}{2}\right) + 1, \sqrt{q}+1\right\}, \sqrt{q}-1\right\}, \tag{3.12}$$

where `round` is the rounding operation to the nearest integer.

As discussed in Example 3.3, the reduction in decoding complexity of the Alamouti space-time block code follows directly from the presence of zero and real elements in the $\mathbf{R}$ matrix. More formally, the number and the location of the zero and the real elements in

the $\mathbf{R}$ matrix can potentially lead to a reduction in decoding complexity by exploiting one or more of the following three properties

1. Some groups of symbols can be decoded separately from other groups.
2. The real and the imaginary parts of a group can be decoded separately.
3. After canceling the interference from some group of symbols, some of the remaining symbols are separately decodable from other symbols.

We note that the early designs of space-time block codes focused on exploiting the first and the second properties. In other words, early designs focused on *separable* space-time block codes such that the decoding can be done on independent groups of symbols. The third property has only been recently utilized in decoding and designing space-time block codes [7][8]. In fact, one contribution of this thesis is to exploit the third property to show that the golden code has a decoding complexity of $\mathcal{O}(q^{2.5})$. We next discuss an example highlighting some different forms of the $\mathbf{R}$ matrix, and show how the three properties can be used to reduce the worst-case decoding complexity.

Example 3.4. Consider the $\mathbf{R}$ matrix in the QR decomposition of seven different 4×4 effective channel matrices as shown in Table 3.1. The matrix $\mathbf{R}$ takes the form

$$\mathbf{R} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}, \tag{3.13}$$

where

- $\mathbf{A} = \begin{bmatrix} r_{1,1} & r_{1,2} \\ 0 & r_{2,2} \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} r_{1,3} & r_{1,4} \\ r_{2,3} & r_{2,4} \end{bmatrix}$ and $\mathbf{D} = \begin{bmatrix} r_{3,3} & r_{3,4} \\ 0 & r_{4,4} \end{bmatrix}$
- The matrices $\mathbf{A}$ and $\mathbf{D}$ are triangular by construction with real diagonal entries. Specifically, $r_{k,k} \in \mathbb{R}$ for $k \in \{1, 2, 3, 4\}$.

Assume that the transmitted symbols are drawn from q-ary QAM alphabet. We will discuss the worst-case decoding complexity for the different forms of the $\mathbf{R}$ matrix. First, however, we define the following variables

- $\mathbf{z}_{12} = [z_1, z_2]^\top$ and $\mathbf{z}_{34} = [z_3, z_4]^\top$
- $\mathbf{x}_{12} = [x_1, x_2]^\top$ and $\mathbf{x}_{34} = [x_3, x_4]^\top$

Table 3.1: The **R** Matrix and its Relationship to Worst-Case Decoding Complexity.

R	**A**	**B**	**D**	Worst-Case Complexity
(1)	$r_{1,2} = 0$	$r_{1,3} = 0$ $r_{1,4} = 0$ $r_{2,3} = 0$ $r_{2,4} = 0$	$r_{3,4} = 0$	$\mathcal{O}(1)$
(2)	$r_{1,2} \in \mathbb{R}$	$r_{1,3} = 0$ $r_{1,4} = 0$ $r_{2,3} = 0$ $r_{2,4} = 0$	$r_{3,4} \in \mathbb{R}$	$\mathcal{O}(\sqrt{q})$
(3)	$r_{1,2} \in \mathbb{C}$	$r_{1,3} = 0$ $r_{1,4} = 0$ $r_{2,3} = 0$ $r_{2,4} = 0$	$r_{3,4} \in \mathbb{C}$	$\mathcal{O}(q)$
(4)	$r_{1,2} \in \mathbb{R}$	$r_{1,3} \in \mathbb{R}$ $r_{1,4} \in \mathbb{R}$ $r_{2,3} \in \mathbb{R}$ $r_{2,4} \in \mathbb{R}$	$r_{3,4} \in \mathbb{R}$	$\mathcal{O}(q^{1.5})$
(5)	$r_{1,2} = 0$	$r_{1,3} \in \mathbb{C}$ $r_{1,4} \in \mathbb{C}$ $r_{2,3} \in \mathbb{C}$ $r_{2,4} \in \mathbb{C}$	$r_{3,4} = 0$	$\mathcal{O}(q^2)$
(6)	$r_{1,2} \in \mathbb{R}$	$r_{1,3} \in \mathbb{C}$ $r_{1,4} \in \mathbb{C}$ $r_{2,3} \in \mathbb{C}$ $r_{2,4} \in \mathbb{C}$	$r_{3,4} \in \mathbb{R}$	$\mathcal{O}(q^{2.5})$
(7)	$r_{1,2} \in \mathbb{C}$	$r_{1,3} \in \mathbb{C}$ $r_{1,4} \in \mathbb{C}$ $r_{2,3} \in \mathbb{C}$ $r_{2,4} \in \mathbb{C}$	$r_{3,4} \in \mathbb{C}$	$\mathcal{O}(q^3)$

1. The cost function for the first row of Table 3.1 is given by

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{z} - \mathbf{R}\mathbf{x}\|^2 \\ &= |z_1 - r_{1,1}x_1|^2 + |z_2 - r_{2,2}x_2|^2 + |z_3 - r_{3,3}x_3|^2 + |z_4 - r_{4,4}x_4|^2 \\ &= |z_1^R - r_{1,1}x_1^R|^2 + |z_1^I - r_{1,1}x_1^I|^2 + |z_2^R - r_{2,2}x_2^R|^2 + |z_2^I - r_{2,2}x_2^I|^2 + \\ &\quad |z_3^R - r_{3,3}x_3^R|^2 + |z_3^I - r_{3,3}x_3^I|^2 + |z_4^R - r_{4,4}x_4^R|^2 + |z_4^I - r_{4,4}x_4^I|^2. \end{aligned} \quad (3.14)$$

   Because the only nonzero elements are the diagonal elements, the symbols x_1 through x_4 are separately decodable. Furthermore, since the diagonal elements are real, the real and the imaginary parts of the symbols are also separable. Consequently, the decoding can be done over eight independent groups, each containing 1 PAM symbol. Therefore, the worst-case decoding complexity is $\mathcal{O}(1)$.

2. The cost function for the second row of Table 3.1 is given by

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{z} - \mathbf{R}\mathbf{x}\|^2 \\ &= \|\mathbf{z}_{12} - \mathbf{A}\mathbf{x}_{12}\|^2 + \|\mathbf{z}_{34} - \mathbf{D}\mathbf{x}_{34}\|^2 \\ &= \|\mathbf{z}_{12}^R - \mathbf{A}\mathbf{x}_{12}^R\|^2 + \|\mathbf{z}_{12}^I - \mathbf{A}\mathbf{x}_{12}^I\|^2 + \|\mathbf{z}_{34}^R - \mathbf{D}\mathbf{x}_{34}^R\|^2 + \|\mathbf{z}_{34}^I - \mathbf{D}\mathbf{x}_{34}^I\|^2. \end{aligned} \quad (3.15)$$

   The $\mathbf{B}$ sub-matrix is zero. As a result, the symbols $\{x_1, x_2\}$ are separable from the symbols $\{x_3, x_4\}$. Furthermore, the elements $r_{1,2}$ and $r_{3,4}$ are real and hence, the real and the imaginary components of the symbols $\{x_1, x_2\}$ and $\{x_3, x_4\}$ are separable. The decoding can be done over four independent groups: $\{x_1^R, x_2^R\}$, $\{x_1^I, x_2^I\}$, $\{x_3^R, x_4^R\}$ and $\{x_3^I, x_4^I\}$. To determine the worst-case decoding complexity, consider the decoding of one of those groups, say $\{x_1^R, x_2^R\}$. For every candidate symbol x_2^R, the decoder finds the corresponding candidate x_1^R with a slicer. Since there are $\sqrt{q}$ ways to pick the PAM symbol x_2^R, and the decoding complexity of the x_1^R slicer is $\mathcal{O}(1)$, the worst-case decoding complexity is $\mathcal{O}(\sqrt{q})$.

3. The cost function for the third row of Table 3.1 is given by

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{z} - \mathbf{R}\mathbf{x}\|^2 \\ &= \|\mathbf{z}_{12} - \mathbf{A}\mathbf{x}_{12}\|^2 + \|\mathbf{z}_{34} - \mathbf{D}\mathbf{x}_{34}\|^2. \end{aligned} \quad (3.16)$$

The **B** sub-matrix is zero, and hence, the symbols $\{x_1, x_2\}$ are also separable from the symbols $\{x_3, x_4\}$. However, the elements $r_{1,2}$ and $r_{3,4}$ are complex, and hence, there is no further reduction in complexity. The decoding can be done over two independent groups: $\{x_1, x_2\}$ and $\{x_3, x_4\}$. To determine the worst-case decoding complexity, consider the decoding of $\{x_1, x_2\}$. For every candidate symbol x_2, the decoder finds the corresponding symbol x_1 with a pair of PAM slicers. Since there are q ways to pick the candidate symbol x_2, and the decoding complexity of the x_1^R and x_1^I slicers is $\mathcal{O}(1)$, the worst-case decoding complexity is $\mathcal{O}(q)$.

4. The cost function for the fourth row of Table 3.1 is given by

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{z} - \mathbf{R}\mathbf{x}\|^2 \\ &= \|\mathbf{z}^R - \mathbf{R}\mathbf{x}^R\|^2 + \|\mathbf{z}^I - \mathbf{R}\mathbf{x}^I\|^2. \end{aligned} \tag{3.17}$$

All the elements of the **R** matrix are real, and hence, the real part of the transmitted symbols is separable from the imaginary part. Therefore, the decoding can be done over two independent groups: $\{x_1^R, x_2^R, x_3^R, x_4^R\}$ and $\{x_1^I, x_2^I, x_3^I, x_4^I\}$. To determine the worst-case decoding complexity, consider the decoding of the group $\{x_1^R, x_2^R, x_3^R, x_4^R\}$. For every candidate group of symbols $\{x_2^R, x_3^R, x_4^R\}$, the decoder finds the corresponding symbol x_1^R with a PAM slicer. Since, there are $\sqrt{q} \times \sqrt{q} \times \sqrt{q}$ ways to pick the three PAM symbols $\{x_2^R, x_3^R, x_4^R\}$, and the decoding complexity of the x_1^R slicer is $\mathcal{O}(1)$, the worst-case decoding complexity is $\mathcal{O}(q^{1.5})$.

For matrices (1) through (4), the decoding of the transmitted symbols was done over two or more groups. Specifically, the decoding was done over 8, 4, 2, and 2 groups of symbols for matrices (1), (2), (3), and (4), respectively. For matrices (5) through (7) the decoding cannot be done over independent groups of symbols. A reduction in decoding complexity, however, might still be possible. We next discuss the worst-case decoding complexity for matrices (5), (6), and (7).

5. Let us define the following variables:

$$\bar{\mathbf{x}}_{12} = \begin{bmatrix} x_1 \\ x_2 \\ 0 \\ 0 \end{bmatrix}, \bar{\mathbf{x}}_{34} = \begin{bmatrix} 0 \\ 0 \\ x_3 \\ x_4 \end{bmatrix}, \mathbf{R}_{12} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \text{ and } \mathbf{R}_{34} = \begin{bmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}. \tag{3.18}$$

Then, the cost function for the fifth row in Table 3.1 is given by

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{z} - \mathbf{R}\mathbf{x}\|^2 \\ &= \|\mathbf{z} - \mathbf{R}_{12}\bar{\mathbf{x}}_{12} - \mathbf{R}_{34}\bar{\mathbf{x}}_{34}\|^2. \end{aligned} \tag{3.19}$$

We now explain how the receiver can exploit the fact that $\mathbf{R}_{12}$ is real to reduce the worst-case decoding complexity. Conceptually, the ML decoder can proceed as follows. For each candidate pair $\mathbf{x}_{34} = [x_3, x_4]^\top$ the total cost of (3.19) can be written as

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{v} - \mathbf{R}_{12}\bar{\mathbf{x}}_{12}\|^2 \\ &= \|\mathbf{v}^R - \mathbf{R}_{12}\bar{\mathbf{x}}_{12}^R\|^2 + \|\mathbf{v}^I - \mathbf{R}_{12}\bar{\mathbf{x}}_{12}^I\|^2 \\ &= |v_1^R - r_{1,1}x_1^R|^2 + |v_1^I - r_{1,1}x_1^I|^2 + |v_2^R - r_{2,2}x_2^R|^2 + |v_2^I - r_{2,2}x_2^I|^2. \end{aligned} \tag{3.20}$$

where we have introduced

$$\mathbf{v} = \mathbf{z} - \mathbf{R}_{34}\bar{\mathbf{x}}_{34}. \tag{3.21}$$

In going from (3.19) to (3.20) we exploited the fact that $\mathbf{A}$ is real and diagonal. This is the key step. Therefore, for every such candidate pair $\mathbf{x}_{34} = [x_3, x_4]^\top$, the receiver finds the corresponding symbols x_1^R, x_1^I, x_2^R, and x_2^I separately.

Intuitively, since the element $r_{1,2}$ is zero, we can decode x_1 and x_2 independently after canceling the interference from the symbols x_3 and x_4. In fact, after canceling the interference from x_3 and x_4, the decoding complexity of x_1 and x_2 is $\mathcal{O}(1)$, since the decoding can be done with a slicer over four independent groups: x_1^R, x_1^I, x_2^R, and x_2^I. Therefore, the worst-case decoding complexity is $\mathcal{O}(q^2)$, where the factor q^2 comes from the number of ways to pick the pair $\mathbf{x}_{34} = [x_3, x_4]^\top$.

6. Using the definition of $\bar{\mathbf{x}}_{12}$, $\bar{\mathbf{x}}_{34}$, $\mathbf{R}_{12}$ and $\mathbf{R}_{34}$ from (3.18), the cost function for the sixth row in Table 3.1 is

$$P(\mathbf{x}) = \|\mathbf{z} - \mathbf{R}_{12}\bar{\mathbf{x}}_{12} - \mathbf{R}_{34}\bar{\mathbf{x}}_{34}\|^2. \tag{3.22}$$

For each candidate pair $\mathbf{x}_{34} = [x_3, x_4]^\top$, the total cost of (3.22) can be written as

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{v} - \mathbf{R}_{12}\bar{\mathbf{x}}_{12}\|^2 \\ &= \|\mathbf{v}^R - \mathbf{R}_{12}\bar{\mathbf{x}}_{12}^R\|^2 + \|\mathbf{v}^I - \mathbf{R}_{12}\bar{\mathbf{x}}_{12}^I\|^2, \end{aligned} \tag{3.23}$$

where $\mathbf{v}$ is given by (3.21). Because the element $r_{1,2}$ is real, we can decode $\mathbf{x}_{12}^R = [x_1^R, x_2^R]$ separately from $\mathbf{x}_{12}^I = [x_1^I, x_2^I]$ after canceling the interference from the symbols x_3 and x_4. After canceling the interference from x_3 and x_4, the decoding complexity of $\{x_1, x_2\}$ is $\mathcal{O}(\sqrt{q})$. This is because the decoding can be done over two independent groups: $\{x_1^R, x_2^R\}$ and $\{x_1^I, x_2^I\}$, and the decoding complexity of each group is $\mathcal{O}(\sqrt{q})$. Therefore, the worst-case decoding complexity is $\mathcal{O}(q^2 \times q^{0.5})$, where the factor q^2 comes from the number of ways to pick the pair $\{x_3, x_4\}$ and the factor $q^{0.5}$ comes from the decoding complexity of $\{x_1, x_2\}$, after canceling the interference from x_3 and x_4.

7. Finally, for matrix (7), all the matrix elements are complex, except for the diagonal elements. As a result, there is no reduction in decoding complexity beyond the reduction offered by a practical implementation of a slicer. Specifically, for every candidate group of symbols $\{x_2, x_3, x_4\}$, the decoder finds the symbol x_1 with a slicer. As a result, the worst-case decoding complexity is $\mathcal{O}(q^3)$, where the factor q^3 comes from the number of ways to choose $\{x_2, x_3, x_4\}$. □

3.4 The R *Matrix in the QR Decomposition of the Effective Channel Matrix for Space-Time Block Codes*

We next discuss the general form of the $\mathbf{R}$ matrix for the different families of space-time block codes discussed in Section 2.4. We will then use the general form of the $\mathbf{R}$ matrix to determine the worst-case ML decoding complexity of the different families of space-time block codes. Specifically, we will examine the worst-case ML decoding complexity in relation

to the properties of the $\mathbf{R}$ matrix in the QR decomposition of the effective channel matrix. Using this framework, we will show that the worst-case decoding complexity for the TAST and perfect space-time block codes is less than what is currently reported in the literature.

In the discussion that follows, we will assume that the number of receive antennas is given by $N = \lceil \mathcal{R} \rceil$ to allow for efficient decoding at the receiver. This is because we need to have as many equations as unknowns at the decoder, and there are NT equations and $\mathcal{R}T$ unknowns. Furthermore, we will also assume quasistatic fading. We will address the worst-case ML decoding complexity for time-varying fading channel later in this section.

3.4.1 Orthogonal Space-Time Block Codes

As mentioned in Section 2.4.1, orthogonal designs decouple the ML detection problem such that each transmitted symbol is detected separately. Orthogonal designs induce orthogonality in the effective channel matrix. For the real-valued effective channel matrix, all the columns are orthogonal to each other and have the same norm. Therefore, the resulting $\mathbf{R}$ matrix in the QR decomposition of the real-valued effective channel matrix for the rate-$\mathcal{R}$ orthogonal design of size $T \times M$ is diagonal and has the general form

$$\mathbf{R} = \beta \cdot \mathbf{I}_{2\mathcal{R}T}, \tag{3.24}$$

where $\beta = \sqrt{\sum_{m=1}^{M} |h_{m,1}|^2}$. The ML cost function can then be written as

$$\begin{aligned} P(\check{\mathbf{x}}) &= \|\check{\mathbf{y}} - \check{\mathbf{H}}\check{\mathbf{x}}\|^2 \\ &= \|\mathbf{z} - \mathbf{R}\check{\mathbf{x}}\|^2 \\ &= \sum_{k=1}^{2\mathcal{R}T} |z_k - \beta \check{x}_k|^2, \end{aligned} \tag{3.25}$$

where $\mathbf{z} = \mathbf{Q}^\top \check{\mathbf{x}}$. From (3.25), we see that the decoding of orthogonal space-time block codes simplifies into the separate decoding of $2\mathcal{R}T$ real symbols. Since each symbol $\check{x}_k$ in (3.25) can be decoded with a slicer, the worst-case decoding complexity is $\mathcal{O}(1)$.

Although orthogonal space-time block codes are trivially decodable without resorting to a QR decomposition, we will see later in Chapter 6 that the proposed high-rate embedded orthogonal designs are not separable, and hence, a significant reduction in decoding

complexity is possible by exploiting the special properties of the $\mathbf{R}$ matrix in the QR decomposition of the effective channel matrix.

3.4.2 Diagonal Algebraic Space-Time Block Codes

For the diagonal algebraic space-time block codes, the complex-valued effective channel matrix takes the general form

$$\begin{aligned}\mathbf{H} &= \operatorname{diag}([h_{1,1},\ldots,h_{M,1}])\mathbf{G} \\ &= \underbrace{\operatorname{diag}([e^{i\angle h_{1,1}},\ldots,e^{i\angle h_{M,1}}])}_{\mathbf{H}_\Phi}\underbrace{\operatorname{diag}([|h_{1,1}|,\ldots,|h_{M,1}|])}_{\mathbf{H}_A}\mathbf{G}. \end{aligned} \tag{3.26}$$

Because $\mathbf{H}_\Phi$ is unitary, the ML decoder cost function in (3.1) can be written as

$$\begin{aligned}P(\mathbf{x}) &= \|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2 \\ &= \|\mathbf{H}_\Phi^*\mathbf{y}-\underbrace{\mathbf{H}_A\mathbf{G}}_{\mathbf{QR}}\mathbf{x}\|^2 \\ &= \|\mathbf{Q}^*\mathbf{H}_\Phi^*\mathbf{y}-\mathbf{R}\mathbf{x}\|^2 \\ &= \|\mathbf{z}_\Phi-\mathbf{R}\mathbf{x}\|^2, \end{aligned} \tag{3.27}$$

where $\mathbf{H}_A\mathbf{G}=\mathbf{Q}\mathbf{R}$ and $\mathbf{z}_\Phi=\mathbf{Q}^*\mathbf{H}_\Phi^*\mathbf{y}$. Any reduction in worst-case ML decoding complexity depends on the properties of the $\mathbf{R}$ matrix in the QR decomposition of $\mathbf{H}_A\mathbf{G}$.

If the rotation matrix $\mathbf{G}$ is complex, then the product $\mathbf{H}_A\mathbf{G}$ is complex, with no orthogonality between the columns, in general. The worst-case ML decoding complexity is $\mathcal{O}\left(q^{M-1}\right)$. This is because there are q^{M-1} ways to choose the first $M-1$ symbols, and for each choice, the last symbol can be decoded with a slicer.

If the rotation matrix $\mathbf{G}$ is real, then the product $\mathbf{H}_A\mathbf{G}$ is also real. Therefore, the resulting $\mathbf{R}$ matrix is

$$\mathbf{R}=\begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,M} \\ & r_{2,2} & \cdots & r_{2,M} \\ & & \ddots & \vdots \\ & & & r_{M,M} \end{bmatrix}, \tag{3.28}$$

where all the elements are real. The ML cost function in (3.27) can then be written as

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{z}_\Phi - \mathbf{R}\mathbf{x}\|^2 \\ &= \|\mathbf{z}_\Phi^R - \mathbf{R}\mathbf{x}^R\|^2 + \|\mathbf{z}_\Phi^I - \mathbf{R}\mathbf{x}^I\|^2. \end{aligned} \tag{3.29}$$

As can be seen from (3.29), the real and imaginary components of the transmitted information symbols can be decoded separately. Since the decoding can be done over two groups, each containing M real PAM symbols, the worst-case decoding complexity is $\mathcal{O}\left(\sqrt{q}^{M-1}\right) = \mathcal{O}\left(q^{\frac{M-1}{2}}\right)$.

3.4.3 Quasiorthogonal Space-Time Block Codes

For quasiorthogonal space-time codes with M transmit antennas, the effective channel matrix is constructed recursively from the effective channel matrix for $\frac{\bar{M}}{2}$ antennas, where $\bar{M}$ is given by in (2.25). This recursive construction is similar to the recursive construction of the quasiorthogonal space-time code itself in (2.24). The complex-valued effective channel matrix is then given by

$$\mathbf{H}_{Quasi}(h_{1,1}, \cdots, h_{\bar{M},1}) = \begin{bmatrix} \mathbf{H}_{Quasi}(h_{1,1}, \cdots, h_{\frac{\bar{M}}{2},1}) & \mathbf{H}_{Quasi}(h_{\frac{\bar{M}}{2}+1,1}, \cdots, h_{\bar{M},1}) \\ \mathbf{H}_{Quasi}(h^*_{\frac{\bar{M}}{2}+1}, \cdots, h^*_{\bar{M},1}) & -\mathbf{H}_{Quasi}(h^*_{1,1}, \cdots, h^*_{\frac{\bar{M}}{2},1}) \end{bmatrix} \mathbf{\Phi}, \tag{3.30}$$

where

- $\mathbf{\Phi} = \begin{bmatrix} e^{\phi_1} & 0 & \cdots & 0 \\ 0 & e^{\phi_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\phi_{\bar{M}}} \end{bmatrix}$
- $\mathbf{H}_{Quasi}(h_{a,1}, \cdots, h_{a+\frac{\bar{M}}{2}-1,1})$ is the effective channel matrix for $\frac{\bar{M}}{2}$ transmit antennas in the fading coefficients $h_{a,1}$ through $h_{a+\frac{\bar{M}}{2}-1,1}$.

By definition, $\mathbf{H}_{Quasi}(h) = h$, and the effective channel matrix for two transmit antennas is the effective channel matrix for the Alamouti space-time block code discussed in Example 2.2.

The quasiorthogonal space-time code induces orthogonality in the effective channel matrix. Specifically, the subspace spanned by the columns $\{1, 4, 6, 7, \cdots\}$ is orthogonal to the subspace spanned by the columns $\{2, 3, 5, 8, \cdots\}$ in the effective channel matrix $\mathbf{H}$. This quasiorthogonal property is easily verified from (3.30).

Let us introduce the permutation matrix $\Pi = [\mathbf{e}_1, \mathbf{e}_4, \cdots, \mathbf{e}_2, \mathbf{e}_3, \cdots]$, then the ML decoder cost function in (3.1) can be written as

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \\ &= \|\mathbf{y} - \mathbf{H}\Pi\Pi^\top\mathbf{x}\|^2 \\ &= \|\mathbf{y} - \bar{\mathbf{H}}\bar{\mathbf{x}}\|^2, \end{aligned} \tag{3.31}$$

where

- $\bar{\mathbf{H}} = \mathbf{H}\Pi = [\bar{\mathbf{h}}_1, \bar{\mathbf{h}}_2, \cdots, \bar{\mathbf{h}}_{\bar{M}}]$ is the permuted effective channel matrix
- $\bar{\mathbf{x}} = \Pi^\top\mathbf{x} = [\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_{\bar{M}}]$ is the permuted transmitted information symbols.

The $\mathbf{R}$ matrix in the QR decomposition of $\bar{\mathbf{H}}$ is then given by

$$\mathbf{R} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,\frac{\bar{M}}{2}} & 0 & \cdots & 0 \\ & \ddots & \vdots & 0 & \ddots & 0 \\ & & r_{\frac{\bar{M}}{2},\frac{\bar{M}}{2}} & 0 & \cdots & 0 \\ & & & r_{1,1} & \cdots & r_{1,\frac{\bar{M}}{2}} \\ & & & & \ddots & \vdots \\ & & & & & r_{\frac{\bar{M}}{2},\frac{\bar{M}}{2}} \end{bmatrix}, \tag{3.32}$$

where all the elements are complex in general except for the diagonal elements. By examining the $\mathbf{R}$ matrix in (3.32), it is evident that quasiorthogonal space-time codes can be decoded over two groups, each having $\frac{\bar{M}}{2} = 2^{\lceil \log_2 M \rceil - 1}$ complex information symbols for $\bar{M}$ antennas. Specifically, the ML cost function in (3.31) can then be written as the sum of two independent cost functions as follows

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{z} - \mathbf{R}\bar{\mathbf{x}}\|^2 \\ &= \|\mathbf{z}_{1,\frac{\bar{M}}{2}} - \mathbf{A}\bar{\mathbf{x}}_{1,\frac{\bar{M}}{2}}\|^2 + \|\mathbf{z}_{\frac{\bar{M}}{2}+1,\bar{M}} - \mathbf{A}\bar{\mathbf{x}}_{\frac{\bar{M}}{2}+1,\bar{M}}\|^2, \end{aligned} \tag{3.33}$$

where $\mathbf{z}_{1,\frac{\bar{M}}{2}} = \left[z_1, \cdots, z_{\frac{\bar{M}}{2}}\right]$, $\mathbf{z}_{\frac{\bar{M}}{2}+1,\bar{M}} = \left[z_{\frac{\bar{M}}{2}+1}, \cdots, z_{\bar{M}}\right]$, $\bar{\mathbf{x}}_{1,\frac{\bar{M}}{2}} = \left[\bar{x}_1, \cdots, \bar{x}_{\frac{\bar{M}}{2}}\right]$, and $\bar{\mathbf{x}}_{\frac{\bar{M}}{2}+1,\bar{M}} = \left[\bar{x}_{\frac{\bar{M}}{2}+1}, \cdots, \bar{x}_{\bar{M}}\right]$. Since the decoding can be done over two groups, each containing $2^{\lceil \log_2 M \rceil - 1}$ complex information symbols, the worst-case ML decoding complexity is $\mathcal{O}\left(q^{2^{\lceil \log_2 M \rceil - 1}-1}\right)$.

3.4.4 Single-Symbol Decodable Space-Time Block Codes

The general form of the $\mathbf{R}$ matrix in the QR decomposition of the effective channel matrix for the different single-symbol decodable codes, after an appropriate permutation, is given by

$$\mathbf{R} = \begin{bmatrix} r_{1,1} & r_{1,2} & 0 & 0 \\ & r_{2,2} & 0 & 0 \\ & & r_{3,3} & r_{3,4} \\ & & & r_{4,4} \end{bmatrix}, \tag{3.34}$$

where all the elements are real. For example, the permutation matrix for [61] and [58] is $\Pi = [\mathbf{e}_1, \mathbf{e}_3, \mathbf{e}_2, \mathbf{e}_4]$. The $\mathbf{R}$ matrix is identical in form to matrix (2) in Example 3.4. Therefore, single-symbol decodable codes require the joint detection of two $\sqrt{q}$-PAM symbols. Consequently, the worst-case ML decoding complexity is $\mathcal{O}(\sqrt{q})$ as discussed in Example 3.4.

3.4.5 Semi-Orthogonal Algebraic Space-Time Block Codes

The complex-valued effective channel matrix for the semi-orthogonal algebraic space-time block codes with M transmit antennas takes the general form

$$\begin{aligned}
\mathbf{H} &= \begin{bmatrix} \text{diag}\left[h_{1,1}, h_{2,1}\cdots, h_{\frac{M}{2},1}\right] & \text{diag}\left[h_{\frac{M}{2}+1,1}, h_{\frac{M}{2}+2,1}\cdots, h_{M,1}\right] \\ \text{diag}\left[h^*_{\frac{M}{2}+1,1}, h^*_{\frac{M}{2}+2,1}\cdots, h^*_{M,1}\right] & -\text{diag}\left[h^*_{1,1}, h^*_{2,1}\cdots, h^*_{\frac{M}{2},1}\right] \end{bmatrix} \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix} \\
&= \underbrace{\begin{bmatrix}
h_{1,1} & 0 & \cdots & 0 & h_{\frac{M}{2}+1,1} & 0 & \cdots & 0 \\
0 & h_{2,1} & \cdots & 0 & 0 & h_{\frac{M}{2}+2,1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & h_{\frac{M}{2},1} & 0 & 0 & \cdots & h_{M,1} \\
h^*_{\frac{M}{2}+1,1} & 0 & \cdots & 0 & -h^*_{1,1} & 0 & \cdots & 0 \\
0 & h^*_{\frac{M}{2}+2,1} & \cdots & 0 & 0 & -h^*_{2,1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & h^*_{M,1} & 0 & 0 & \cdots & -h^*_{\frac{M}{2},1}
\end{bmatrix}}_{\bar{\mathbf{H}}} \underbrace{\begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix}}_{\bar{\mathbf{G}}}.
\end{aligned} \tag{3.35}$$

In order to determine the general form of the matrix $\mathbf{R}$, we first perform a QR decomposition on the matrix $\bar{\mathbf{H}}$, such that $\bar{\mathbf{H}} = \bar{\mathbf{Q}}\bar{\mathbf{R}}$. Because the matrix $\bar{\mathbf{H}}$ is orthogonal, we have

$$\begin{aligned}
\bar{\mathbf{H}} &= \bar{\mathbf{Q}} \begin{bmatrix}
r_{1,1} & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & r_{\frac{M}{2},\frac{M}{2}} & 0 & \cdots & 0 \\
0 & \cdots & 0 & r_{1,1} & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & 0 & \cdots & r_{\frac{M}{2},\frac{M}{2}}
\end{bmatrix} \\
&= \underbrace{\bar{\mathbf{Q}} \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}}_{\bar{\mathbf{R}}},
\end{aligned} \tag{3.36}$$

where $\mathbf{D} = \text{diag}\left(\left[r_{1,1}, \cdots, r_{\frac{M}{2},\frac{M}{2}}\right]\right)$. Substituting $\bar{\mathbf{H}} = \bar{\mathbf{Q}}\bar{\mathbf{R}}$ into (3.35) yields

$$\begin{aligned}
\mathbf{H} &= \bar{\mathbf{Q}}\bar{\mathbf{R}}\bar{\mathbf{G}} \\
&= \bar{\mathbf{Q}} \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix} \\
&= \bar{\mathbf{Q}} \begin{bmatrix} \mathbf{DG} & \mathbf{0} \\ \mathbf{0} & \mathbf{DG} \end{bmatrix}. \qquad (3.37)
\end{aligned}$$

We perform one more QR decomposition on the matrix product $\mathbf{DG}$ such that $\mathbf{DG} = \bar{\bar{\mathbf{Q}}}\bar{\bar{\mathbf{R}}}$, and we obtain the desired QR decomposition of $\mathbf{H} = \mathbf{QR}$ where

$$\mathbf{Q} = \bar{\mathbf{Q}} \begin{bmatrix} \bar{\bar{\mathbf{Q}}} & \mathbf{0} \\ \mathbf{0} & \bar{\bar{\mathbf{Q}}} \end{bmatrix} \qquad (3.38)$$

and

$$\mathbf{R} = \begin{bmatrix} \bar{\bar{\mathbf{R}}} & \mathbf{0} \\ \mathbf{0} & \bar{\bar{\mathbf{R}}} \end{bmatrix} = \begin{bmatrix}
r_{1,1} & \cdots & r_{1,\frac{M}{2}} & 0 & \cdots & 0 \\
 & \ddots & \vdots & 0 & \ddots & 0 \\
 & & r_{\frac{M}{2},\frac{M}{2}} & 0 & \cdots & 0 \\
 & & & r_{1,1} & \cdots & r_{1,\frac{M}{2}} \\
 & & & & \ddots & \vdots \\
 & & & & & r_{\frac{M}{2},\frac{M}{2}}
\end{bmatrix}, \qquad (3.39)$$

where all the elements are real since both $\mathbf{D}$ and $\mathbf{G}$ are real. The $\mathbf{R}$ matrix in (3.39) has the same form as the $\mathbf{R}$ matrix for quasiorthogonal space-time block codes in (3.32). The difference, however, is that all the elements in (3.39) are real. The ML decoder cost function in (3.1) can be written as

$$\begin{aligned}
P(\mathbf{x}) &= \|\mathbf{y} - \mathbf{Hx}\|^2 \\
&= \|\mathbf{z} - \mathbf{Rx}\|^2 \\
&= \|\mathbf{z}_{1,\frac{M}{2}} - \bar{\bar{\mathbf{R}}}\mathbf{x}_{1,\frac{M}{2}}\|^2 + \|\mathbf{z}_{\frac{M}{2}+1,M} - \bar{\bar{\mathbf{R}}}\mathbf{x}_{\frac{M}{2}+1,M}\|^2 \\
&= \|\mathbf{z}^R_{1,\frac{M}{2}} - \bar{\bar{\mathbf{R}}}\mathbf{x}^R_{1,\frac{M}{2}}\|^2 + \|\mathbf{z}^I_{1,\frac{M}{2}} - \bar{\bar{\mathbf{R}}}\mathbf{x}^I_{1,\frac{M}{2}}\|^2 + \\
&\quad \|\mathbf{z}^R_{\frac{M}{2}+1,M} - \bar{\bar{\mathbf{R}}}\mathbf{x}^R_{\frac{M}{2}+1,M}\|^2 + \|\mathbf{z}^I_{\frac{M}{2}+1,M} - \bar{\bar{\mathbf{R}}}\mathbf{x}^I_{\frac{M}{2}+1,M}\|^2, \qquad (3.40)
\end{aligned}$$

where $\mathbf{z}_{a,b} = [z_a, \cdots, z_b]^\top$ and $\mathbf{x}_{a,b} = [x_a, \cdots, x_b]^\top$.

As can be seen from (3.40), SAST codes can be decoded over four groups of symbols, each has $\frac{M}{2}$ real symbols drawn from $\sqrt{q}$-PAM alphabet. Hence, the worst-case ML decoding complexity is $\mathcal{O}\left(\sqrt{q}^{\frac{M}{2}-1}\right) = \mathcal{O}\left(q^{\frac{M/2-1}{2}}\right)$.

3.4.6 Threaded Algebraic Space-Time Block Codes

For the TAST code matrix in (2.31) and the complex-valued system model in (2.6), the effective channel matrix of the rate-$\mathcal{R}$ TAST code with M transmit and N receive antennas is

$$\begin{aligned}
\mathbf{H} &= \begin{bmatrix} \mathtt{diag}\left(\mathbf{J}^0\mathbf{h}_1\right) & \mathtt{diag}\left(\mathbf{J}^1\mathbf{h}_1\right) & \cdots & \mathtt{diag}\left(\mathbf{J}^{\mathcal{R}-1}\mathbf{h}_1\right) \\ \mathtt{diag}\left(\mathbf{J}^0\mathbf{h}_2\right) & \mathtt{diag}\left(\mathbf{J}^1\mathbf{h}_2\right) & \cdots & \mathtt{diag}\left(\mathbf{J}^{\mathcal{R}-1}\mathbf{h}_2\right) \\ \vdots & \vdots & \ddots & \vdots \\ \mathtt{diag}\left(\mathbf{J}^0\mathbf{h}_N\right) & \mathtt{diag}\left(\mathbf{J}^1\mathbf{h}_N\right) & \cdots & \mathtt{diag}\left(\mathbf{J}^{\mathcal{R}-1}\mathbf{h}_N\right) \end{bmatrix} (\mathbf{I}_{\mathcal{R}} \otimes \mathbf{G}) \\
&= \begin{bmatrix} \mathtt{diag}\left(\mathbf{J}^0\mathbf{h}_1\right)\mathbf{G} & \mathtt{diag}\left(\mathbf{J}^1\mathbf{h}_1\right)\mathbf{G} & \cdots & \mathtt{diag}\left(\mathbf{J}^{\mathcal{R}-1}\mathbf{h}_1\right)\mathbf{G} \\ \mathtt{diag}\left(\mathbf{J}^0\mathbf{h}_2\right)\mathbf{G} & \mathtt{diag}\left(\mathbf{J}^1\mathbf{h}_2\right)\mathbf{G} & \cdots & \mathtt{diag}\left(\mathbf{J}^{\mathcal{R}-1}\mathbf{h}_2\right)\mathbf{G} \\ \vdots & \vdots & \ddots & \vdots \\ \mathtt{diag}\left(\mathbf{J}^0\mathbf{h}_N\right)\mathbf{G} & \mathtt{diag}\left(\mathbf{J}^1\mathbf{h}_N\right)\mathbf{G} & \cdots & \mathtt{diag}\left(\mathbf{J}^{\mathcal{R}-1}\mathbf{h}_N\right)\mathbf{G} \end{bmatrix}, \qquad (3.41)
\end{aligned}$$

where

- $\mathbf{J} = [\phi\mathbf{e}_M, \mathbf{e}_1, \cdots, \mathbf{e}_{M-1}]$
- $\mathbf{h}_n = [h_{1,n}, h_{2,n}, \cdots, h_{M,n}]^\top$, $n \in \{1, \cdots, N\}$
- diag($\mathbf{a}$) is the diagonal matrix with the vector $\mathbf{a}$ on the diagonal
- $\mathbf{G}$ is the $M \times M$ generator matrix.

For $\mathcal{R} > 1$, the TAST code is not separable [4]. However, a reduction in worst-case ML decoding complexity is possible with real generator matrices, as we discuss in the following theorem.

Theorem 3.1. *The worst-case ML decoding complexity of a rate-$\mathcal{R}$ TAST code for M transmit antennas and with real generator matrices is $\mathcal{O}(q^{(\mathcal{R}-\frac{1}{2})M-\frac{1}{2}})$.*

Proof. The worst-case decoding complexity of the rate-$\mathcal{R}$ TAST code is the worst-case decoding complexity of the first $\mathcal{R}-1$ threads, multiplied by the worst-case decoding complexity of the last thread. Consider the worst-case decoding complexity of a rate-one TAST code (i.e., last thread), which is obtained by setting $\mathcal{R}=1$ in (2.31), for q-ary QAM alphabet. As discussed earlier, the rate-one TAST code is simply the DAST code in (2.23). Therefore, the worst-case decoding complexity of the rate-one TAST code is the same as the DAST code, namely $\mathcal{O}\left(q^{\frac{M-1}{2}}\right)$. Therefore, the worst-case decoding of the rate-$\mathcal{R}$ TAST code is $\mathcal{O}\left(q^{(\mathcal{R}-1)M}\right)\times\mathcal{O}\left(q^{\frac{M-1}{2}}\right)$, where the first term is the worst-case decoding complexity of the first $\mathcal{R}-1$ threads, and the last term is the decoding complexity of the last thread. Consequently, the worst-case decoding complexity is $\mathcal{O}\left(q^{(\mathcal{R}-\frac{1}{2})M-\frac{1}{2}}\right)$. □

The reduction in worst-case ML decoding complexity can be derived by considering the properties of the $\mathbf{R}$ matrix in the QR decomposition of the effective channel matrix in (3.41), which has the general form

$$\mathbf{R}=\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{0} & \mathbf{D}\end{bmatrix}, \tag{3.42}$$

where

- $\mathbf{A}$ is an $M\times M$ real-valued upper triangular matrix
- $\mathbf{B}$ is an $(\mathcal{R}-1)M\times(\mathcal{R}-1)M$ complex-valued matrix.
- $\mathbf{D}$ is an $M\times M$ upper triangular matrix.

The fact that $\mathbf{A}$ is real follows directly from the fact that it represents the $\mathbf{R}$ matrix in the QR decomposition of the first M columns of the effective channel matrix in (3.41). The first M columns of the TAST effective channel matrix represent the effective channel matrix of a DAST code with M transmit antennas and N receive antennas. As discussed in Section 3.4.2, the $\mathbf{R}$ matrix is real-valued when the generator matrix is real, and hence, the matrix $\mathbf{A}$ is real.

Let us define $\mathbf{R}_a$ and $\mathbf{R}_b$ as follows

$$\mathbf{R}_a = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \text{ and } \mathbf{R}_b = \begin{bmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}. \tag{3.43}$$

Then, the ML cost function in (3.1) can be written as

$$P(\mathbf{x}) = \|\mathbf{z} - \mathbf{R}_a\mathbf{x}_a - \mathbf{R}_b\mathbf{x}_b\|^2, \tag{3.44}$$

where $\mathbf{x}_a = [x_1, \cdots, x_M, 0, \cdots, 0]^\top$ and $\mathbf{x}_b = [0, \cdots, 0, x_{M+1}, \cdots, x_{\mathcal{R}M}]^\top$. The ML decoder proceeds as follows. For each candidate group of symbols $\{x_{M+1}, \cdots, x_{\mathcal{R}M}\}$, the total cost of (3.44) can be written as

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{v} - \mathbf{R}_a\mathbf{x}_a\|^2 \\ &= \|\mathbf{v}^R - \mathbf{R}_a\mathbf{x}_a^R\|^2 + \|\mathbf{v}^I - \mathbf{R}_a\mathbf{x}_a^I\|^2, \end{aligned} \tag{3.45}$$

where $\mathbf{v} = \mathbf{z} - \mathbf{R}_b\mathbf{x}_b$.

Because the matrix $\mathbf{R}_a$ is real, we can decode $\{x_1^R, \cdots, x_M^R\}$ separately from $\{x_1^I, \cdots, x_M^I\}$ in the absence of interference from the symbols x_{M+1} through $x_{\mathcal{R}M}$. The decoding complexity of $\{x_1, \cdots, x_M\}$ is $\mathcal{O}\left(\sqrt{q}^{M-1}\right) = \mathcal{O}\left(q^{\frac{M-1}{2}}\right)$. Therefore, the worst-case ML decoding complexity for the rate-$\mathcal{R}$ TAST code is $\mathcal{O}\left(q^{(\mathcal{R}-1)M}\right) \times \mathcal{O}\left(q^{\frac{M-1}{2}}\right)$, where the first term is the worst-case ML decoding complexity of the symbols x_{M+1} through $x_{\mathcal{R}M}$ and the second term is the worst-case ML decoding complexity of the symbols x_1 through x_M.

3.4.7 Perfect Space-Time Block Codes

The effective channel matrix of the perfect code is identical in form to the effective channel matrix of the TAST code in (3.41). Therefore, The worst-case decoding complexity of the perfect space-time codes with real generator matrices for QAM alphabet is the same as that of the TAST code; namely $\mathcal{O}(q^{(\mathcal{R}-\frac{1}{2})M-\frac{1}{2}})$. The proof follows identically to Theorem 3.1. For HEX alphabet, wherein the real and imaginary parts of the alphabet itself are not separable, there is no reduction in decoding complexity beyond that offered by an efficient implementation of a slicer. Hence, the worst-case decoding complexity is $\mathcal{O}(q^{\mathcal{R}M-1})$.

3.5 Summary of Previous Space-Time Block Codes

In Table 3.2, we compare orthogonal, quasiorthogonal, diagonal algebraic, semi-orthogonal algebraic, single-symbol decodable, threaded algebraic, and perfect space-time block codes in terms of their code rate $\mathcal{R}$, delay T, number of real symbols in each group that are jointly detected, and worst-case ML decoding complexity. The following assumptions were made in comparing these space-time block codes:

- The comparison assumes quasistatic fading channel. This is because, on time-varying channels, the different families of space-time block codes lose their reduced complexity decoding, if any, with the exception of the diagonal algebraic space-time block codes with real rotation matrices. For orthogonal, quasiorthogonal, semi-orthogonal and single-symbol decodable codes, the reduction in decoding complexity in quasistatic fading is due to the orthogonality of some columns in the effective channel matrix. The orthogonality property, however, is lost in time-varying fading. For the diagonal algebraic space-time block codes with real rotation matrices, a reduction in decoding complexity is possible because the complex-valued channel matrix can be transformed into a real-valued channel matrix as discussed in Section 3.4.2. This property holds in both quasistatic and time-varying fading.

- The complex information symbols are assumed to be drawn from a q-ary QAM alphabet.

- The comparison is for an arbitrary number of antennas M, except for the single-symbol decodable codes, where $M \in \{3, 4\}$, and except for perfect codes, where $M \neq 3$ and $M \neq 6$ since the modulation alphabet is HEX, wherein the real and imaginary parts are not separable;

- We assumed real rotation matrices for the diagonal algebraic, threaded algebraic, and perfect space-time block codes.

Table 3.2: Comparison of Different Space-Time Block Codes in Terms of Rate, Delay, Number of Real Symbols Detected Per Group, and Decoding Complexity.

Space-Time Code	Rate ($\mathcal{R}$)	Delay (D_{min})	Number of Real Symbols Per Group	Decoding Complexity
Orthogonal	$\frac{\lceil M/2\rceil+1}{2\lceil M/2\rceil}$	$\begin{cases} \frac{1}{2\mathcal{R}}\begin{pmatrix} M \\ \lceil M/2\rceil \end{pmatrix} & M=4k \\ \frac{1}{\mathcal{R}}\begin{pmatrix} M \\ \lceil M/2\rceil \end{pmatrix} & M\neq 4k \end{cases}$	1	$\mathcal{O}(1)$
Quasi-orthogonal	1	$2^{\lceil \log_2 M\rceil}$	$2^{\lceil \log_2 M\rceil}$	$\mathcal{O}\left(q^{2^{\lceil \log_2 M\rceil -1}-1}\right)$
DAST	1	M	M	$\mathcal{O}\left(q^{\frac{M-1}{2}}\right)$
SSD	1	$2\lceil \frac{M}{2}\rceil$	2	$\mathcal{O}\left(\sqrt{q}\right)$
SAST	1	M	$\lceil \frac{M}{2}\rceil$	$\mathcal{O}\left(q^{\frac{\lceil M/2\rceil -1}{2}}\right)$
TAST	$\{1,\cdots,M\}$	M	$2\mathcal{R}M$	$\mathcal{O}\left(q^{(\mathcal{R}-\frac{1}{2})M-\frac{1}{2}}\right)$
Perfect	$\{1,\cdots,M\}$	M	$2\mathcal{R}M$	$\mathcal{O}\left(q^{(\mathcal{R}-\frac{1}{2})M-\frac{1}{2}}\right)$

3.6 Conclusions

We presented a unified framework for determining the worst-case ML decoding complexity of space-time block codes. The framework is based on the observation that the space-time block code induces structure in its effective channel matrix. Reduced complexity

decoding is then possible by exploiting the properties of the upper-triangular matrix in the QR decomposition of the effective channel matrix. Using the framework, we determined the worst-case ML decoding complexity for orthogonal, quasiorthogonal, diagonal algebraic, single-symbol decodable, semi-orthogonal algebraic, thread algebraic and perfect space-time block codes.

Using the framework presented in this chapter, we will show in Chapter 4 that the worst-case decoding complexity of the golden code is significantly less than that of an exhaustive-search ML decoder. Furthermore, the framework will be used to determine the decoding complexity of the proposed asymmetric golden code in Chapter 5, and the proposed embedded orthogonal space-time block codes in Chapter 6.

CHAPTER 4

A FAST DECODING ALGORITHM FOR THE GOLDEN CODE

Of particular interest in mobile applications are MIMO systems with two transmit and two receive antennas; this configuration is able to provide both diversity and multiplexing gains while keeping costs at a minimum. The golden code is a space-time code for two transmit and two receive antennas that was proposed independently in [7] and [8]. The golden code has a nonvanishing determinant and hence, achieves the full diversity-multiplexing frontier of Zheng and Tse [3]. Furthermore, it performs better than previously reported full-rate codes with two transmit antennas in terms of the SNR required to achieve a target error probability.

The golden code comes in three variations: the Belfiore-Rekaya-Viterbo golden code [7], the Dayal-Varanasi golden code [8], and the WiMAX golden code [10]. These variations are isomorphic, in the sense that one can be transformed into another by multiplying on the left and right by unitary matrices [8]. Since the determinant is invariant to such transformations, all three variations have identical rate, diversity, and coding gain. Furthermore, we will show that they all have the same decoding complexity. For the sake of concreteness, we present our results in the context of the Dayal-Varanasi golden code [8], and then show that they are in fact applicable to all three variations [7][8][10].

Because each golden code codeword conveys four information symbols from a q-ary QAM alphabet, the complexity of an exhaustive-search decoder is proportional to q^4. For example, the authors in [18] noted that the worst-case ML decoding complexity of the golden code grows with the fourth-power of the signal constellation size. In this chapter we prove that the golden code with q-ary QAM is *fast decodable*, by which we mean that ML decoding is possible with a worst-case decoding complexity of only $\mathcal{O}(q^{2.5})$. The golden code is fast decodable regardless of whether the channel varies with time. We also present an efficient implementation of a fast decoder that has low average complexity.

The remainder of the chapter is organized as follows. In Section 4.1, we review the construction of the golden code and prove that it is fast decodable. In Section 4.2 we introduce a new fast ML decoder for the golden code that has low average complexity. In Section 4.3, we show that the fast ML decoder applies to other variations of the golden code. In Section 4.4, we compare the average complexity of the proposed detector to a conventional golden code detector. We conclude the chapter in Section 4.5.

4.1 The Golden Code is Fast Decodable

In this section, we first review the construction of the golden code. Then, we study the properties of the $\mathbf{R}$ matrix in the QR decomposition of the effective channel matrix that lead to a reduction in the worst-case ML decoding complexity. Finally, we show that the golden is *fast decodable*, a property that we define as follows:

Definition 4.1. A rate-two space-time block for the two-input two-output channel is said to be *fast decodable* if the worst-case ML decoding complexity is $\mathcal{O}(q^{2.5})$.

4.1.1 The Golden Code Induces Structure in Effective Channel

The golden code transmits four complex information symbols over two symbol periods, so that the rate is two symbols per signaling interval. In particular, the Dayal-Varanasi golden code encodes one pair of information symbols $\mathbf{a} = [x_1, x_2]^\top$ onto the main diagonal of the code matrix, and it encodes a second pair of symbols $\mathbf{b} = [x_3, x_4]^\top$ onto the off-diagonal, yielding [8]:

$$\mathbf{C} = \begin{bmatrix} \tilde{a}_1 & 0 \\ 0 & \tilde{a}_2 \end{bmatrix} + \phi \begin{bmatrix} 0 & \tilde{b}_1 \\ \tilde{b}_2 & 0 \end{bmatrix} \tag{4.1}$$

where:

$$\tilde{\mathbf{a}} = \mathbf{G}\mathbf{a}, \quad \tilde{\mathbf{b}} = \mathbf{G}\mathbf{b}, \quad \mathbf{G} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix},$$
$$\theta = \frac{1}{2}\tan^{-1}(2), \quad \phi = e^{j\pi/4}. \tag{4.2}$$

Substituting the definition of the golden code from (4.1) and (4.2) into (2.2), the vector of samples $\mathbf{y} = [y_1[1], y_1[2], y_2[1], y_2[2]]^\top$ received at a receiver with two antennas at the two

time instances can be written as the output of an effective four-input four-output channel:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{4.3}$$

where $\mathbf{x} = [x_1, \ldots, x_4]^\top$ is the vector of information symbols, $n = [n_1[1], \ldots, n_2[2]]^\top$ is the noise, and where $\mathbf{H} = \bar{\mathbf{H}}\Psi$ is the *effective channel matrix*:

$$\mathbf{H} = \underbrace{\begin{bmatrix} h_{1,1}[1] & 0 & \phi h_{2,1}[1] & 0 \\ 0 & h_{2,1}[2] & 0 & \phi h_{1,1}[2] \\ h_{1,2}[1] & 0 & \phi h_{2,2}[1] & 0 \\ 0 & h_{2,2}[2] & 0 & \phi h_{1,2}[2] \end{bmatrix}}_{\bar{\mathbf{H}}} \underbrace{\begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}}_{\Psi}, \tag{4.4}$$

where $c = \cos(\theta)$, $s = \sin(\theta)$, $\phi = e^{j\pi/4}$, and $\theta = \frac{1}{2}\tan^{-1}(2)$.

The structure of the golden code induces special properties in this effective matrix that we exploit to reduce decoding complexity. The following lemma examines these special properties of the $\mathbf{R}$ matrix in the orthogonal-triangular (QR) decomposition of the effective channel matrix $\mathbf{H}$ in (4.4).

Lemma 4.1. *(The Key Property): The* $\mathbf{R}$ *matrix in a QR decomposition* $\mathbf{H} = \mathbf{QR}$ *of the effective channel (4.4) has the form*

$$\mathbf{R} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}, \tag{4.5}$$

where both of the upper triangular matrices $\mathbf{A}$ *and* $\mathbf{D}$ *are entirely real.*

Proof. See Appendix B. □

A few remarks:

- Both $\mathbf{A} = \begin{bmatrix} r_{1,1} & r_{1,2} \\ 0 & r_{2,2} \end{bmatrix}$ and $\mathbf{D} = \begin{bmatrix} r_{3,3} & r_{3,4} \\ 0 & r_{4,4} \end{bmatrix}$ are triangular by construction with real diagonal entries, so the key property is essentially the fact that both $r_{1,2}$ and $r_{3,4}$ are real.

- To demonstrate that $r_{1,2} = \mathbf{h}_1^* \mathbf{h}_2 / \|\mathbf{h}_1\|$ is real, it is sufficient to show that the inner product between the first two columns is real, a fact which is easily verified by direct computation:

$$\begin{aligned} \mathbf{h}_1^* \mathbf{h}_2 &= \cos(\theta)\sin(\theta)(|h_{1,1}[1]|^2 - |h_{2,1}[2]|^2 + |h_{1,2}[1]|^2 - |h_{2,2}[2]|^2) \\ &= \frac{1}{\sqrt{5}}(|h_{1,1}[1]|^2 - |h_{2,1}[2]|^2 + |h_{1,2}[1]|^2 - |h_{2,2}[2]|^2). \end{aligned} \tag{4.6}$$

- The lemma applies regardless of whether the channel is quasistatic or time-varying.

- The submatrix $\mathbf{B}$ is not mentioned because all four of its entries are generally complex.

- The fact that $r_{1,2}$ is real enables the decoder in the next section to reduce both the worst-case decoding complexity and the average decoding complexity. In contrast, the fact that $r_{3,4}$ is real enables only a reduction in average complexity. It has no impact on the worst-case complexity.

- The $\mathbf{R}$ matrix in (4.5) has the same form as matrix (6) in Example 3.4. Therefore, we expect a reduction in decoding complexity such that the worst-case ML decoding complexity is $\mathcal{O}(q^{2.5})$. We next discuss how a receiver decides on the vector $\mathbf{x}$ such that the worst-case complexity is $\mathcal{O}(q^{2.5})$, and we introduce some intermediate variables that will be used later when discussing a fast ML decoder with low average complexity.

4.1.2 The Golden Code is Fast Decodable

We now show how the key property of Lemma 4.1 enables fast decoding. If we define $\mathbf{z}_{12} = [z_1, z_2]^\top$ and $\mathbf{z}_{34} = [z_3, z_4]^\top$, where $\mathbf{z} = \mathbf{Q}^* \mathbf{y}$, then the ML decision minimizes the cost function

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \|\mathbf{z} - \mathbf{R}\mathbf{x}\|^2 \\ &= \|\mathbf{z}_{12} - \mathbf{A}\mathbf{a} - \mathbf{B}\mathbf{b}\|^2 + \|\mathbf{z}_{34} - \mathbf{D}\mathbf{b}\|^2. \end{aligned} \tag{4.7}$$

The last equality follows from (4.5). Therefore, the ML decisions $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ can be found recursively using:

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b} \in \mathcal{A}^2} \{\|\mathbf{z}_{12} - \mathbf{A}\mathbf{a}_*(\mathbf{b}) - \mathbf{B}\mathbf{b}\|^2 + \|\mathbf{z}_{34} - \mathbf{D}\mathbf{b}\|^2\}, \tag{4.8}$$

$$\hat{\mathbf{a}} = \mathbf{a}_*(\hat{\mathbf{b}}), \tag{4.9}$$

where

$$\mathbf{a}_*(\mathbf{b}) = \arg\min_{\mathbf{a}\in\mathcal{A}^2}\{\|\mathbf{z}_{12} - \mathbf{A}\mathbf{a} - \mathbf{B}\mathbf{b}\|^2\}. \tag{4.10}$$

The function $\mathbf{a}_*(\mathbf{b})$ in (4.10) can be viewed as producing the best $\mathbf{a}$ for a given $\mathbf{b}$. With this interpretation, the optimization in (4.8) can be viewed as that of finding the best $\mathbf{b}$ when $\mathbf{a}$ is optimized.

The optimization (4.10) is equivalent to ML detection for a channel $\mathbf{A}$ with an input of $\mathbf{a}$ and an output:

$$\mathbf{v} = \mathbf{z}_{12} - \mathbf{B}\mathbf{b}. \tag{4.11}$$

It can be solved by a sphere detector applied to a two-level tree. With two QAM inputs and without any constraints on $\mathbf{A}$, the worst-case complexity would be $\mathcal{O}(q)$. This is because for every candidate symbol x_2, the decoder decides on the corresponding symbol x_1 with a slicer. But the golden code induces the special property that $\mathbf{A}$ is real, which enables us to determine the real components of $\mathbf{a}$ *independently* from its imaginary components in (4.10). Specifically, we may rewrite (4.10) as:

$$\begin{aligned}
\mathbf{a}_*(\mathbf{b}) &= \arg\min_{\mathbf{a}\in\mathcal{A}^2}\{\|\mathbf{v}^R - \mathbf{A}\mathbf{a}^R\|^2 + \|\mathbf{v}^I - \mathbf{A}\mathbf{a}^I\|^2\} && (4.12)\\
&= \arg\min_{\mathbf{a}^R\in(\mathcal{A}^R)^2}\{\|\mathbf{v}^R - \mathbf{A}\mathbf{a}^R\|^2\} + j\cdot\arg\min_{\mathbf{a}^I\in(\mathcal{A}^I)^2}\{\|\mathbf{v}^I - \mathbf{A}\mathbf{a}^I\|^2\}. && (4.13)
\end{aligned}$$

Thus, the optimization in (4.10) decomposes into the pair of independent optimizations of (4.13). Since each optimization in (4.13) is equivalent to ML detection for a real channel with two $\sqrt{q}$-PAM inputs, each has a worst-case complexity of $\mathcal{O}(\sqrt{q})$. Thus, the overall complexity of (4.13) is $\mathcal{O}(\sqrt{q})$. We thus arrive at our main theorem.

Theorem 4.1. *(Golden Code is Fast Decodable): A maximum-likelihood decoder for the golden code with a q-ary QAM alphabet can be implemented with a worst-case complexity of* $\mathcal{O}(q^{2.5})$.

Proof. As described in (4.8), the ML decision can be found by stepping through each of the q^2 candidate values for $\mathbf{b}$, and for each implement the $\mathcal{O}(\sqrt{q})$ optimization of (4.13). □

4.2 A Fast ML Decoder With Low Average Complexity

The decoding strategy used to prove the fast-decodable theorem has a low worst-case complexity but a high average complexity. In this section we present an efficient implementation of an ML decoder for the golden code that has both low average complexity and a worst-case complexity of $\mathcal{O}(q^{2.5})$.

A conventional sphere decoder for the golden code is based on a four-level tree, with a different x_i associated with each level. In contrast, as illustrated in Figure 4.1, we propose a four-level tree that associates $\mathbf{b}^R = (x_3^R, x_4^R)$ with the first level, $\mathbf{b}^I = (x_3^I, x_4^I)$ with the second level, $\mathbf{a}^R = (x_1^R, x_2^R)$ with the third level, and $\mathbf{a}^I = (x_1^I, x_2^I)$ with the fourth level. This new tree is a direct result of the fact that $\mathbf{A}$ and $\mathbf{D}$ are real (Lemma 4.1), which allows us to rewrite the ML cost function from (4.7) as

$$P(\mathbf{x}) = \underbrace{\|\mathbf{v}^I - \mathbf{A}\mathbf{a}^I\|^2}_{P_1} + \underbrace{\|\mathbf{v}^R - \mathbf{A}\mathbf{a}^R\|^2}_{P_2} + \underbrace{\|\mathbf{z}_{34}^I - \mathbf{D}\mathbf{b}^I\|^2}_{P_3} + \underbrace{\|\mathbf{z}_{34}^R - \mathbf{D}\mathbf{b}^R\|^2}_{P_4}. \tag{4.14}$$

Thus, as illustrated in Figure 4.1, (4.14) shows that the total cost of a leaf node $\mathbf{x}$ decomposes into the sum $\sum_i P_i$ of four branch metrics, where P_i denotes the branch metric for a branch at the $(4-i)$-th stage of the tree.

Besides inducing a new tree structure, the fact that $\mathbf{D}$ is real also leads to a significant reduction in the complexity of the Schnorr-Euchner sorting for the first two stages of the tree. Specifically, the fact that $\mathbf{D}$ is real leads to a second-stage branch metric P_3 that is *independent* of the starting node ($\mathbf{b}^R$). Therefore, we can perform a single sort for the symbol pair ($\mathbf{b}^R$) emanating from the root, and *simultaneously* a single sort for the symbol pair ($\mathbf{b}^I$) emanating from its children.

The pseudocode of an efficient implementation of the proposed ML golden code detector is shown in Figure 4.2. The first five lines represent initializations. In particular, the first two lines are a QR decomposition of the effective channel matrix in (4.4) and the computation of $\mathbf{z}$ in (4.7). The squared sphere radius $\hat{P}$, which represents the smallest cost (4.14) encountered so far, is initialized to infinity to ensure ML decoding (line 3). Sorting or Schnorr-Euchner enumeration is used for faster convergence. Only two sorting operations (line 4 and line 5) are required. In the pseudocode, the complex QAM alphabet

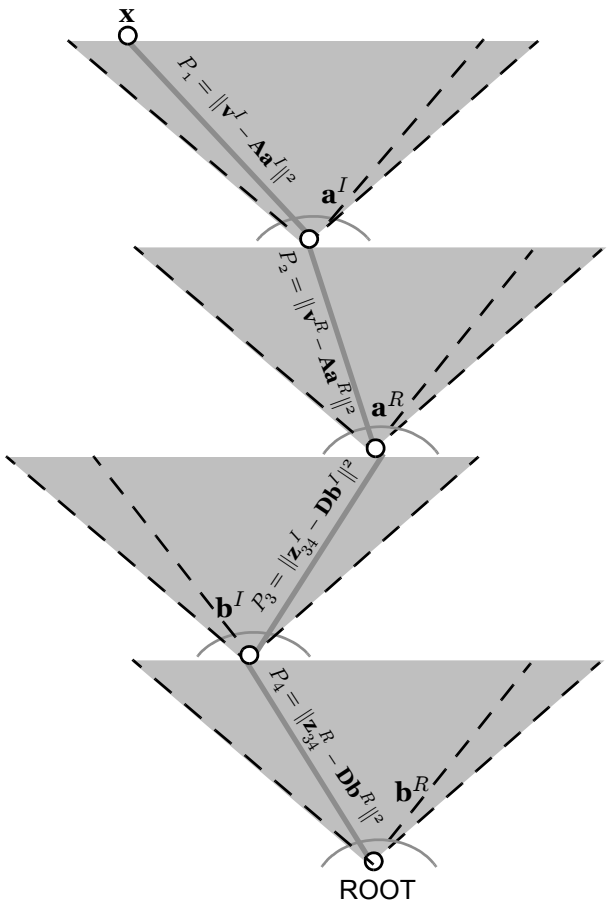


Figure 4.1: The structure of the proposed detection tree and its branch metrics. The cost function for the leaf node is the sum of the branch metrics, $P(\mathbf{x}) = P_1 + P_2 + P_3 + P_4$.

$[\mathbf{Q}, \mathbf{R}] = \texttt{QR decomposition}(\mathbf{H})$
$\mathbf{z} = \mathbf{Q}^{*}\mathbf{y}$
$\hat{P} = \infty$
$[P_4, \Pi_4] = \texttt{sort}_{a \in \mathcal{A}}((z_3^R - r_{3,3}a^R - r_{3,4}a^I)^2 + (z_4^R - r_{4,4}a^I)^2)$
$[P_3, \Pi_3] = \texttt{sort}_{a \in \mathcal{A}}((z_3^I - r_{3,3}a^R - r_{3,4}a^I)^2 + (z_4^I - r_{4,4}a^I)^2)$
for k from 1 to q **do**
  if $(P_4(k) + P_3(1)) > \hat{P}$ **then**
    break
  end
  for l from 1 to q **do**
    if $(P_3(l) + P_4(k)) > \hat{P}$ **then**
      break
    end
    $[x_3^R, x_3^I, x_4^R, x_4^I] = [\mathcal{A}(\Pi_4(k))^R, \mathcal{A}(\Pi_3(l))^R, \mathcal{A}(\Pi_4(k))^I, \mathcal{A}(\Pi_3(l))^I]$
    $v_1 = z_1 - r_{1,3}x_3 - r_{1,4}x_4$
    $v_2 = z_2 - r_{2,3}x_3 - r_{2,4}x_4$
    $\hat{P}_1 = \hat{P}_2 = \hat{P}$
    $\mathcal{X} = \texttt{list}(v_2^R / r_{2,2})$
    for m from 1 to $\sqrt{q}$ **do**
      $P_2 = (v_2^R - r_{2,2}\mathcal{X}(m))^2$
      if $P_2 > \hat{P}_2$ **then, break, end**
      $u_1^R = v_1^R - r_{1,2}\mathcal{X}(m)$, $w = Q(u_1^R / r_{1,1})$, $P_2 = (u_1^R - r_{1,1}w)^2 + P_2$
      if $P_2 < \hat{P}_2$ **then**
        $x_1^R = w$, $x_2^R = \mathcal{X}(m)$, $\hat{P}_2 = P_2$
      end
    end
    $\mathcal{X} = \texttt{list}(v_2^I / r_{2,2})$
    for n from 1 to $\sqrt{q}$ **do**
      $P_1 = (v_2^I - r_{2,2}\mathcal{X}(n))^2$
      if $P_1 > \hat{P}_1$ **then, break, end**
      $u_1^I = v_1^I - r_{1,2}\mathcal{X}(n)$, $w = Q(u_1^I / r_{1,1})$, $P_1 = (u_1^I - r_{1,1}w)^2 + P_1$
      if $P_1 < \hat{P}_1$ **then**
        $x_1^I = w$, $x_2^I = \mathcal{X}(n)$, $\hat{P}_1 = P_1$
      end
    end
    $P = \hat{P}_1 + \hat{P}_2 + P_3(l) + P_4(k)$
    if $P < \hat{P}$ **then**
      $\hat{\mathbf{x}} = [x_1, x_2, x_3, x_4]$, $\hat{P} = P$
    end
  end
end

Figure 4.2: Pseudocode of a fast ML decoder for the golden code.

$\mathcal{A}$ is represented by an ordered list, so that $\mathcal{A}(k)$ indexes the k-th symbol in the list.

The remainder of algorithm can be interpreted as a two-level *complex* sphere decoder to choose the symbol pair $\mathbf{b} = (x_3, x_4)^\top$, followed by an independent pair of two-level *real* sphere decoders that separately decode $\mathbf{a}^R = (x_1^R, x_2^R)^\top$ and $\mathbf{a}^I = (x_1^I, x_2^I)^\top$.

The two-level complex sphere decoder incorporates two common optimizations: radius update (line 38) and pruning (line 7, line 11). While these optimizations do not affect the worst-case decoding complexity, they affect the average complexity significantly. The first level of the complex sphere decoder considers candidate pairs $\mathbf{b}^R$ in ascending order of their branch metric P_4 (line 6). The second level of the complex sphere decoder considers candidate pairs $\mathbf{b}^I$ in ascending order of their branch metric P_3 (line 10). After forming $\mathbf{b} = [x_3, x_4]^\top$ (line 14), the decoder removes the interference caused by $\mathbf{b}$ and forms the two intermediate variables v_1 and v_2 of (4.11), which are functions of the symbols x_1 and x_2 only (line 15 and line 16). Following the two-level complex sphere decoder and interference cancelation, the decoder decides on the symbol pairs $\mathbf{a}^R$ and $\mathbf{a}^I$ separately using an independent pair of two-level real sphere decoders.

The function `list` is used to implement sorting for the final two stages of the tree; it returns a list of candidate symbols drawn from the $\sqrt{q}$-ary PAM alphabet $\mathcal{A}^R$, sorted in ascending order of distance to the input argument. As described in [65], it can be implemented efficiently using a table lookup.

After initializing the sphere radius for decoding $\mathbf{a}^R = (x_1^R, x_2^R)^\top$ (line 17) and forming the sorted list of best candidate symbols (line 18), the real sphere decoder chooses the symbol x_2^R that has the lowest branch metric P_2 (line 20). The interference from the symbol x_2^R is then subtracted, and a decision is made on the symbol x_1^R using the PAM slicer $Q(\ \cdot\ )$ (line 22); the slicer function $Q(x)$ returns the symbol from the PAM alphabet $\mathcal{A}^R$ that is closest to x and is given in (3.12). The branch-metric P_2 for the current candidate symbol pair $\mathbf{a}^R$ is also computed in line 22, and radius update occurs if it is less than the previous smallest value $\hat{P}_2$ (line 24). The real sphere decoder includes pruning and radius update (line 21 and line 24, respectively).

Decoding the symbol pair $\mathbf{a}^I$ follows identically to the decoding of the symbol pair $\mathbf{a}^R$

and is shown in line 27 through line 35. Importantly, $\mathbf{a}^R$ and $\mathbf{a}^I$ are decoded independently. Therefore, although the pseudocode shows a serial implementation that decodes first $\mathbf{a}^R$ and second $\mathbf{a}^I$, a hardware implementation could decode them simultaneously (in parallel), thus decreasing decoding latency. The overall cost P for the current candidate symbol vector is updated in line 44. Radius update and best candidate vector update occurs if the current cost P is less than the previous smallest cost $\hat{P}$ (line 38).

The algorithm could be embellished to further reduce average complexity. For example, the columns of $\mathbf{H}$ could be permuted using a BLAST ordering. (In this case the only permutations for which the key property of Lemma 4.1 will still hold are $\{1,2,3,4\}$, $\{1,2,4,3\}$, $\{2,1,3,4\}$, $\{2,1,4,3\}$, $\{3,4,1,2\}$, $\{3,4,2,1\}$, $\{4,3,1,2\}$ and $\{4,3,2,1\}$.) For the sake of clarity of exposition, however, we have chosen not to include such refinements in Figure 4.2. Such refinements, which have no effect on the worst-case complexity, are well-known in the literature and their application to the pseudocode is straightforward.

We remark that a quasistatic channel does not offer any additional reduction in decoding complexity, as compared to a time-varying channel. This is a direct result of the fact that the entries of $\mathbf{B}$ in (4.5) are generally complex, regardless of whether the channel is quasistatic or time-varying.

4.3 Golden Code Variations

In this section we show that the proposed fast ML decoder, although presented in the context of the Dayal-Varanasi version of the golden code [8], is equally applicable to the Belfiore-Rekaya-Viterbo [7] and WiMAX [10] versions of the golden code.

Substituting the definition of the Belfiore-Rekaya-Viterbo golden code from [[7], eqn. (9)] into (2.2), the vector of samples received at a receiver with two antennas will again be

given by (4.3), but with a new effective channel matrix of the form:

$$\mathbf{H}=\begin{bmatrix} h_{1,1}[1] & 0 & h_{2,1}[1] & 0 \\ 0 & h_{2,1}[2] & 0 & ih_{1,1}[2] \\ h_{1,2}[1] & 0 & h_{2,2}[1] & 0 \\ 0 & h_{2,2}[2] & 0 & ih_{1,2}[2] \end{bmatrix} \begin{bmatrix} c-si & 0 & 0 & 0 \\ 0 & s+ci & 0 & 0 \\ 0 & 0 & c-si & 0 \\ 0 & 0 & 0 & s+ci \end{bmatrix} \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}, \tag{4.15}$$

where $c = \cos(\theta)$, $s = \sin(\theta)$, $\phi = e^{j\pi/4}$, and $\theta = \frac{1}{2}\tan^{-1}(2)$. The information symbols $[b, a, d, c]$ in [7] have been relabeled as $[x_1, x_2, x_3, x_4]$ in (4.3). Since $c - si$ and $s + ci$ have unity magnitude, we can transform this effective matrix into the one in (4.4) simply by rotating the channel coefficients $h_{i,j}[k]$. These rotations have no impact on complexity. In particular, the real coefficients in the $\mathbf{R}$ matrix will remain real, even after rotation. Therefore, the proposed fast ML decoder is applicable.

Similarly, substituting the definition of the WiMAX golden code from [[10], Section 8.4.8.3.3] (matrix C) into (2.2) will again yield (4.3), but with a new effective channel matrix of the form:

$$\mathbf{H} = \begin{bmatrix} h_{1,1}[1] & 0 & h_{2,1}[1] & 0 \\ 0 & -ih_{2,1}[2] & 0 & -h_{1,1}[2] \\ h_{1,2}[1] & 0 & h_{2,2}[1] & 0 \\ 0 & -ih_{2,2}[2] & 0 & -h_{1,2}[2] \end{bmatrix} \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}. \tag{4.16}$$

The information symbol vector $[S_k, iS_{k+3}, S_{k+1}, -S_{k+2}]$ in [10] has been relabeled to [x_1, x_2, x_3, x_4] in (4.3). Just as before, this effective matrix differs from that in (4.4) only by the rotated channel coefficients. Therefore, the proposed fast ML decoder is again applicable to this version of the golden code.

4.4 Numerical Results

In Figure 4.3 we compare the average complexity of the proposed fast ML decoder to a conventional ML decoder. Unlike our previous discussion where complexity is quantified by the number of leaf nodes visited, complexity here is quantified by the average number of nodes visited while searching the tree. The channel was modeled using (2.2) with quasistatic i.i.d. Rayleigh fading, with constant coefficients within each codeword block, but

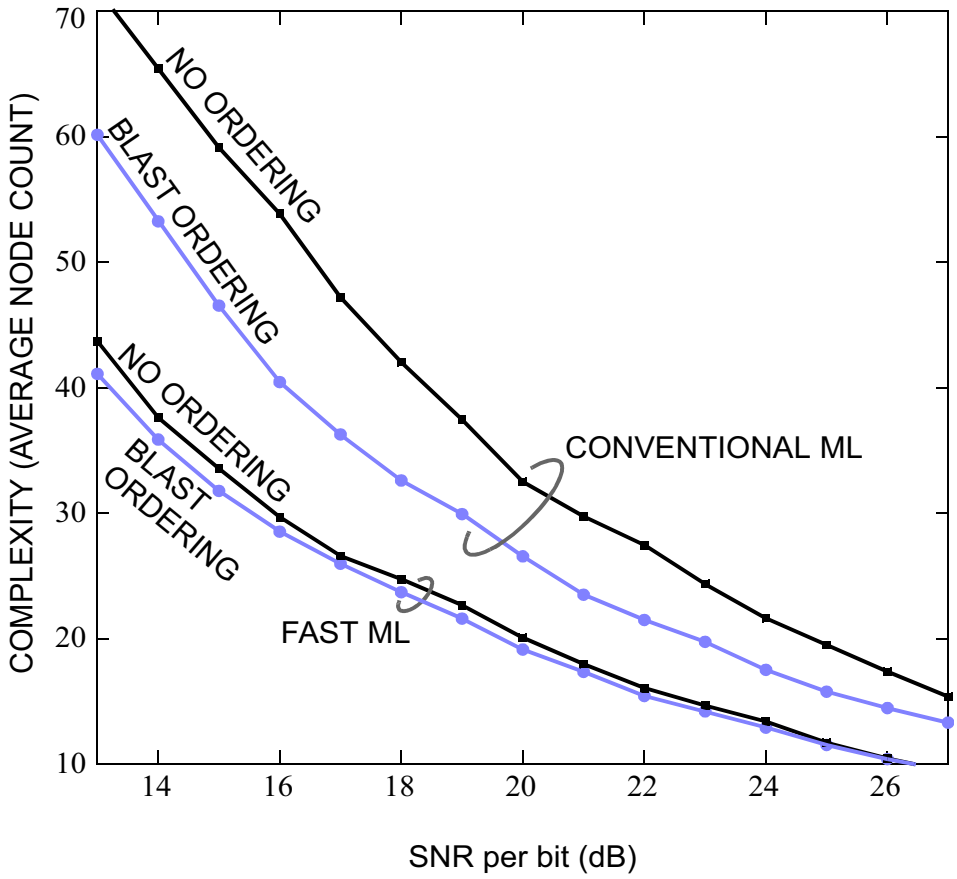


Figure 4.3: Average decoding complexity versus SNR for golden code with 64-QAM.

independent fading from block to block. The alphabet was 64-QAM. The fast ML decoder was implemented following the pseudocode of Figure 4.2. The conventional ML decoder was implemented using an efficient four-level complex sphere decoder with Schnorr-Euchner enumeration. Results are shown for two cases of channel matrix column ordering: no ordering and BLAST reordering.

As can be seen from Figure 4.3, with no column ordering, the proposed fast ML decoder is about 45% less complex than a conventional ML decoder. With BLAST ordering, the proposed ML decoder is about 30% less complex than a conventional decoder. Beyond the advantages shown in Figure 4.3, the proposed algorithm has three additional advantages that are not reflected in Figure 4.3, namely:

- the proposed algorithm reduces the number of Schnorr-Euchner sort operations for the first two stages to only two, compared with a conventional decoder that can require

as many as $q+1$.

- the proposed algorithm can avoid BLAST ordering without a high complexity penalty.
- decoding of the symbol pairs $\mathbf{a}^R$ and $\mathbf{a}^I$ can be done in parallel, reducing decoding latency.

We remark that for 64-QAM, the worst-case ML decoding complexity of the proposed decoder is $\mathcal{O}(64^{2.5}) = 32768$, and the worst-case ML decoding complexity of a conventional decoder is $\mathcal{O}(q^3) = 262144$. The reduction in worst-case ML decoding complexity of the proposed decoder is 98% compared to a conventional decoder. The reduction in average complexity, however, is about 30% with ordering. This result is not surprising given the fact that the proposed decoder visits fewer nodes in the upper part of the tree search, rather than the lower part of the tree. Furthermore, for the conventional decoder, pruning of the child nodes occurs if the sum of the branch metrics up to the parent node exceeds the sphere radius value. Since the sum of the branch metrics increases as we ascend the tree, we are more likely to prune branches in the higher layers in the tree. Therefore, we can expect that the average complexity gap between the conventional decoder and proposed decoder to be less than the worst-case complexity for the two reasons mentioned above, namely that the conventional decoder is more likely to prune branches in the upper levels and because the proposed decoder visits fewer nodes in the upper two levels of the tree, but not the lower two levels.

4.5 Conclusions

The golden code induces special structure in the effective channel matrix. By recognizing and exploiting this structure we have proven that the worst-case complexity of an ML decoder for the golden code with q-ary QAM is $\mathcal{O}(q^{2.5})$, regardless of whether the channel varies with time. By further exploiting this structure we have proposed a fast ML decoding algorithm based on a unique tree construction that outperforms a conventional ML detector on four fronts simultaneously: worst-case complexity, average complexity, sorting complexity, and decoding latency.

CHAPTER 5

THE ASYMMETRIC GOLDEN CODE

The desire for high spectral efficiency for the practically important system having two transmit and two receive antennas motivated the construction of high-rate space-time block codes. Due to the perceived high complexity of the golden code, a class of fast-decodable space-time codes for the two-input two-output channel was recently proposed as alternatives to the golden code [16][17][18]. We call the codes of [16][17][18] *overlaid Alamouti codes* because they are based on a layering of two Alamouti space-time block codes. See [24] for a survey of the overlaid Alamouti codes. We have shown in Chapter 4 that for the case of QAM alphabets, the golden code itself is fast-decodable on both quasistatic and time-varying channels with worst-case decoding complexity of $\mathcal{O}(q^{2.5})$. The overlaid Alamouti codes have a worst-case decoding complexity that is even smaller, namely $\mathcal{O}(q^2)$.

The overlaid Alamouti codes lose their fast-decodable property when the channel varies with time, a common occurrence in wireless applications with high mobility. In this chapter we propose a full-rate space-time code with a nonvanishing determinant that is fast decodable in both quasistatic and rapidly time-varying channels. In particular, we propose the *asymmetric golden code*, which is constructed over QAM alphabets, and in which one layer of the golden code is scaled with respect to the other. We prove that its worst-case decoding complexity is $\mathcal{O}(q^{2.5})$, regardless of whether the channel is time varying. For the special case of quasistatic fading, we prove that it has a decoding complexity of $\mathcal{O}(q^2)$, the same as the overlaid Alamouti codes. Furthermore, we present an efficient implementation of a fast ML decoder for quasistatic channels. The asymmetric golden code has the lowest decoding complexity among all previously proposed codes and maintains its low decoding complexity on both quasistatic and time-varying channels.

The remainder of the chapter is organized as follows. In Section 5.1, we present a unified framework for the encoding of high-rate space-time block codes for the two-input two-output

channel. In Section 5.2 we present the proposed asymmetric golden code and prove its fast decoding properties in quasistatic and time-varying channels. In Section 5.3, we introduce a fast ML decoder with low average complexity for quasistatic fading. In Section 5.4 we present numerical results, and in Section 5.5 we draw conclusions.

5.1 A Unified Framework for the Encoding of High-Rate Space-Time Codes for the Two-Input Two-Output Channel

We present a common framework for comparing the golden code and the overlaid Alamouti codes in a unified way. In Section 5.2 we will use this framework to describe the asymmetric golden code. All of these codes transmit four complex information symbols $\{x_1, x_2, x_3, x_4\}$ from two transmit antennas in two signaling intervals, and they may all be viewed as the sum of a pair of rate-one codes. In particular, all may be described by the following 2×2 space-time code:

$$\mathbf{C}(\mathbf{x}) = \mathbf{C}_1(\mathbf{a}) + \mathbf{\Phi}_L \mathbf{C}_1(\hat{\mathbf{b}}) \mathbf{\Phi}_R, \tag{5.1}$$

where:

- $\mathbf{x} = [x_1, x_2, x_3, x_4]^\top$, $\mathbf{a} = [x_1, x_2]^\top$, $\mathbf{b} = [x_3, x_4]^\top$, $\hat{\mathbf{b}} = \mathbf{\Phi}_P \mathbf{b}$;
- $\mathbf{\Phi}_P$ is a unitary precoding matrix;
- $\mathbf{C}_1(\cdot)$ is a rate-one encoder for symbol pairs $\mathbf{a}$ or $\hat{\mathbf{b}}$;
- $\mathbf{\Phi}_L$ and $\mathbf{\Phi}_R$ are 2×2 matrices that multiply $\mathbf{C}_1(\hat{\mathbf{b}})$ on the left and right, respectively.

The four parameters $\mathbf{C}_1(\cdot)$, $\mathbf{\Phi}_L$, $\mathbf{\Phi}_R$, and $\mathbf{\Phi}_P$ are chosen to ensure full diversity, maximize the coding gain and reduce decoding complexity.

One possibility for $\mathbf{C}_1(\cdot)$ is the diagonal algebraic space-time code of [27], defined by:

$$\mathbf{C}_{DAST}(a_1, a_2) = \begin{bmatrix} u_1 & 0 \\ 0 & u_2 \end{bmatrix}, \tag{5.2}$$

where

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \mathbf{G} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \ \mathbf{G} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \tag{5.3}$$

and

$$c = \cos(\theta), s = \sin(\theta), \text{ and } \theta = \frac{1}{2}\tan^{-1}(2). \tag{5.4}$$

Another possibility is the Alamouti code:

$$\mathbf{C}_{Alamouti}(a_1, a_2) = \frac{1}{\sqrt{2}} \begin{bmatrix} a_1 & a_2 \\ -a_2^* & a_1^* \end{bmatrix}, \tag{5.5}$$

where the constant $1/\sqrt{2}$ ensures that the total transmit energy is identical to the energy of the underlying alphabet.

The golden code is a special case of (5.1) with:

$$\mathbf{C}_1 = \mathbf{C}_{DAST}, \mathbf{\Phi}_P = \mathbf{I}_2, \mathbf{\Phi}_L = e^{i\pi/4}\mathbf{I}_2, \mathbf{\Phi}_R = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{5.6}$$

The overlaid Alamouti codes of [16][17] are a special case of (5.1) with:

$$\mathbf{C}_1 = \mathbf{C}_{Alamouti}, \mathbf{\Phi}_P = \begin{bmatrix} \phi_1 & \phi_2 \\ -\phi_2^* & \phi_1^* \end{bmatrix}, \mathbf{\Phi}_L = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \mathbf{\Phi}_R = \mathbf{I}_2, \tag{5.7}$$

where

$$\phi_1 = \frac{1}{\sqrt{7}}(1+i) \text{ and } \phi_2 = \frac{1}{\sqrt{7}}(1+2i). \tag{5.8}$$

Finally, the overlaid Alamouti code of [18] is also a special case of (5.1), with:

$$\mathbf{C}_1 = \mathbf{C}_{Alamouti}, \mathbf{\Phi}_P = \mathbf{I}_2, \mathbf{\Phi}_L = \frac{\alpha}{|\alpha|} \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}, \mathbf{\Phi}_R = \mathbf{I}_2, \tag{5.9}$$

where

$$\alpha = (1-\sqrt{7}) + i(1+\sqrt{7}). \tag{5.10}$$

We proved in Chapter 4 that the worst-case decoding complexity of the golden code is $\mathcal{O}(q^{2.5})$ in both quasistatic and time-varying fading for QAM alphabets. The overlaid Alamouti codes with QAM are fast-decodable with worst-case decoding complexity of $\mathcal{O}(q^2)$ in quasistatic fading channels [24], but lose their fast decoding property and have a worst-case complexity of $\mathcal{O}(q^3)$ on time-varying channels. We next present the asymmetric golden code, which has a worst-case decoding complexity of $\mathcal{O}(q^2)$ in quasistatic channels and $\mathcal{O}(q^{2.5})$ in time-varying channels, which is the lowest decoding complexity compared to previously reported rate-two space-time block codes for the two-input two-output channel.

5.2 The Asymmetric Golden Code

We propose a novel full-rate full-diversity space-time block code for the two-input two-output channel: the *asymmetric golden code*. We first describe encoding, and then prove the fast decoding properties of the asymmetric golden code.

5.2.1 The Asymmetric Golden Code Encoder

We propose the asymmetric golden code, a special case of (5.1) with

$$\mathbf{C}_1 = \mathbf{C}_{\mathcal{K}}, \mathbf{\Phi}_P = \mathbf{I}_2, \mathbf{\Phi}_L = \mathcal{K}\mathbf{I}_2, \mathbf{\Phi}_R = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{5.11}$$

where $\mathcal{K} \in (0,1)$ is an *asymmetry coefficient*, to be specified later. The factor $\mathcal{K}$ in $\mathbf{\Phi}_L$ ensures that the second layer will be scaled by $\mathcal{K}$ before being added to the first layer. The rate-one encoder $\mathbf{C}_{\mathcal{K}}$ in (5.11) is a modified version of the diagonal algebraic encoder in (5.2), defined by:

$$\mathbf{C}_{\mathcal{K}}(a_1, a_2) = \sqrt{\frac{2}{1+\mathcal{K}^2}} \begin{bmatrix} u_1 & 0 \\ 0 & u_2^* \end{bmatrix}, \tag{5.12}$$

where $[u_1, u_2]^\top$ is given by (5.3).

Comparing (5.12) to $\mathbf{C}_{DAST}$ of (5.2), we see two differences: the constant $\sqrt{\frac{2}{1+\mathcal{K}^2}}$, which ensures that the average transmit energy is equal to that of the underlying alphabet, and the conjugation of u_2. The latter makes $\mathbf{C}_{\mathcal{K}}$ different from $\mathbf{C}_{DAST}$ even when $\mathcal{K} = 1$. Let $\mathbf{a} = [x_1, x_2]^\top$ and $\mathbf{b} = [x_3, x_4]^\top$. Then, in terms of the original information symbols, the asymmetric golden code is given by

$$\mathbf{C} = \sqrt{\frac{2}{1+\mathcal{K}^2}} \left(\begin{bmatrix} \tilde{a}_1 & 0 \\ 0 & \tilde{a}_2^* \end{bmatrix} + \mathcal{K} \begin{bmatrix} 0 & \tilde{b}_1 \\ \tilde{b}_2^* & 0 \end{bmatrix} \right) \tag{5.13}$$

$$= \sqrt{\frac{2}{1+\mathcal{K}^2}} \begin{bmatrix} cx_1 + sx_2 & \mathcal{K}(cx_3 + sx_4) \\ \mathcal{K}(-sx_3^* + cx_4^*) & -sx_1^* + cx_2^* \end{bmatrix}, \tag{5.14}$$

where:

$$\tilde{\mathbf{a}} = \mathbf{G}\mathbf{a}, \quad \tilde{\mathbf{b}} = \mathbf{G}\mathbf{b}, \text{ and } \mathbf{G} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \tag{5.15}$$

and c and s are given by (5.4).

Theorem 5.1. *The asymmetry coefficient $\mathcal{K} \in (0,1)$ that maximizes the coding gain (2.12) for the space-time block code in (5.13) with QAM alphabet is $\mathcal{K} = 1/\sqrt{3}$. The resulting coding gain is $1/\sqrt{20}$.*

Proof. See Appendix C. □

5.2.2 The Effective Channel Matrix and its Key Properties

We next describe the effective channel matrix induced by the asymmetric golden code. We then establish the key properties of this matrix, and describe a maximum-likelihood decoder that exploits the key properties to reduce complexity in rapid time-varying fading as well as quasistatic fading.

Substituting the definition of the asymmetric golden code from (5.13) into (2.2), the vector of samples after conjugating the second and fourth samples received at a receiver with two antennas at the two time instances can be written as the output of an effective four-input four-output channel:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{5.16}$$

where $\mathbf{y} = [y_1[1], y_1^*[2], y_2[1], y_2^*[2]]^\top$ is the vector of received samples after conjugation, $\mathbf{x} = [x_1, x_2, x_3, x_4]^\top$ is the vector of information symbols, $n = [n_1[1], n_1^*[2], n_2[1], n_2^*[2]]^\top$ is the noise, and where $\mathbf{H} = \bar{\mathbf{H}}\Psi$ is the *effective channel matrix*:

$$\mathbf{H} = \sqrt{\frac{3}{2}} \underbrace{\begin{bmatrix} h_{1,1}[1] & 0 & \mathcal{K}h_{2,1}[1] & 0 \\ 0 & h_{2,1}^*[2] & 0 & \mathcal{K}h_{1,1}^*[2] \\ h_{1,2}[1] & 0 & \mathcal{K}h_{2,2}[1] & 0 \\ 0 & h_{2,2}^*[2] & 0 & \mathcal{K}h_{1,2}^*[2] \end{bmatrix}}_{\bar{\mathbf{H}}} \underbrace{\begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}}_{\Psi}. \tag{5.17}$$

Similar to the golden code, the asymmetric golden code induces special properties in its effective channel matrix that we exploit to reduce decoding complexity. The following two lemmas examine the special properties of the $\mathbf{R}$ matrix in the orthogonal-triangular (QR) decomposition of the effective channel matrix $\mathbf{H}$ in (5.17).

Lemma 5.1. *(The Key Property in Time-Varying Fading): The* $\mathbf{R}$ *matrix in a QR decomposition* $\mathbf{H} = \mathbf{QR}$ *of the effective channel (5.17) has the form*

$$\mathbf{R} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}, \tag{5.18}$$

where both of the upper triangular matrices $\mathbf{A}$ *and* $\mathbf{D}$ *are entirely real.*

Proof. The proof follows immediately from the proof of the key property for fast decoding of the golden code (see Appendix B), where $\bar{\mathbf{H}}$ and $\mathbf{\Psi}$ are given in (5.17). □

A few remarks:

- Both $\mathbf{A} = \begin{bmatrix} r_{1,1} & r_{1,2} \\ 0 & r_{2,2} \end{bmatrix}$ and $\mathbf{D} = \begin{bmatrix} r_{3,3} & r_{3,4} \\ 0 & r_{4,4} \end{bmatrix}$ are triangular by construction with real diagonal entries, so the key property is the fact that both $r_{1,2}$ and $r_{3,4}$ are real.

- To demonstrate that $r_{1,2} = \mathbf{h}_1^* \mathbf{h}_2 / \|\mathbf{h}_1\|$ is real, it is sufficient to show that the inner product between the first two columns is real, a fact which is easily verified by direct computation:

$$\begin{aligned} \mathbf{h}_1^* \mathbf{h}_2 &= \frac{3}{2} \cos(\theta) \sin(\theta) (|h_{1,1}[1]|^2 - |h_{2,1}[2]|^2 + |h_{1,2}[1]|^2 - |h_{2,2}[2]|^2) \\ &= \frac{3}{2\sqrt{5}} (|h_{1,1}[1]|^2 - |h_{2,1}[2]|^2 + |h_{1,2}[1]|^2 - |h_{2,2}[2]|^2). \end{aligned} \tag{5.19}$$

- The lemma applies regardless of whether the channel is quasistatic or time-varying. For the case of quasistatic fading, however, permuting the channel matrix prior to QR decomposition leads to an $\mathbf{R}$ matrix with $r_{1,2} = r_{3,4} = 0$. We next discuss the key property of this permuted effective channel matrix in quasistatic fading.

Let us introduce the permutation matrix $\mathbf{\Pi} = [\mathbf{e}_1, \mathbf{e}_4, \mathbf{e}_2, \mathbf{e}_3]$, where $\mathbf{e}_i$ is the i-th column of the 4×4 identity matrix, so that (5.16) can be written as:

$$\begin{aligned} \mathbf{y} &= \mathbf{H\Pi\Pi}^\top \mathbf{x} + \mathbf{n} \\ &= \mathbf{H}_\Pi \mathbf{x}_\Pi + \mathbf{n}, \end{aligned} \tag{5.20}$$

where $\mathbf{x}_\Pi = \mathbf{\Pi}^\top \mathbf{x} = [x_1, x_4, x_2, x_3]^\top$ and $\mathbf{H}_\Pi = \mathbf{H\Pi} = \bar{\mathbf{H}}\mathbf{\Psi\Pi}$ is the *permuted effective channel matrix* given by

$$\mathbf{H}_\Pi = \sqrt{\frac{3}{2}} \underbrace{\begin{bmatrix} h_{1,1}[1] & 0 & \mathcal{K}h_{2,1}[1] & 0 \\ 0 & h^*_{2,1}[2] & 0 & \mathcal{K}h^*_{1,1}[2] \\ h_{1,2}[1] & 0 & \mathcal{K}h_{2,2}[1] & 0 \\ 0 & h^*_{2,2}[2] & 0 & \mathcal{K}h^*_{1,2}[2] \end{bmatrix}}_{\bar{\mathbf{H}}} \underbrace{\begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}}_{\Psi} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{\Pi}. \quad (5.21)$$

The following lemma presents the key property in quasistatic fading.

Lemma 5.2. *(The Key Property in Quasistatic Fading): The* $\mathbf{R}$ *matrix in a QR decomposition* $\mathbf{H}_\Pi = \mathbf{QR}$ *of the permuted effective channel (5.21) has the form*

$$\mathbf{R} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}, \quad (5.22)$$

where both $\mathbf{A}$ *and* $\mathbf{D}$ *are real and diagonal.*

Proof. See Appendix D. □

We also remark that by construction, both $\mathbf{A} = \begin{bmatrix} r_{1,1} & r_{1,2} \\ 0 & r_{2,2} \end{bmatrix}$ and $\mathbf{D} = \begin{bmatrix} r_{3,3} & r_{3,4} \\ 0 & r_{4,4} \end{bmatrix}$ are upper triangular with real diagonal entries, so the key property in quasistatic fading is essentially the fact that $r_{1,2} = r_{3,4} = 0$.

5.2.3 The Asymmetric Golden Code is Fast-Decodable

We start with the case of time-varying fading. We show how the key property of Lemma 5.2 enables fast decoding. If we define $\mathbf{z}_{12} = [z_1, z_2]^\top$ and $\mathbf{z}_{34} = [z_3, z_4]^\top$, where $\mathbf{z} = \mathbf{Q}^*\mathbf{y}$, then the ML decision minimizes the cost function

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{y} - \mathbf{Hx}\|^2 \\ &= \|\mathbf{z} - \mathbf{Rx}\|^2 \\ &= \|\mathbf{z}_{12} - \mathbf{Aa} - \mathbf{Bb}\|^2 + \|\mathbf{z}_{34} - \mathbf{Db}\|^2. \end{aligned} \quad (5.23)$$

The last equality follows from (5.22). The ML minimization in (5.23) for the asymmetric golden code is identical to the ML minimization in (4.7) for the golden code. Therefore,

the ML decisions $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ can be found recursively using (4.8) and (4.9), respectively. As discussed in Theorem 4.1, the worst-case ML decoding complexity for finding the decisions $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ is $\mathcal{O}(q^{2.5})$. This leads us to the fast-decodable theorem for the asymmetric golden code in time-varying fading.

Theorem 5.2. *(Asymmetric Golden Code is Fast-Decodable on Time-Varying Fading): A maximum-likelihood decoder for the asymmetric golden code with a q-ary QAM alphabet can be implemented with a worst-case complexity of $\mathcal{O}(q^{2.5})$ on time-varying fading channel.*

Proof. The proof follows immediately from Theorem 4.1. Specifically, the ML decision can be found by stepping through each of the q^2 candidate values for $\mathbf{b}$ as described in (4.8), and for each implement the $\mathcal{O}(\sqrt{q})$ optimization of (4.13). □

We next discuss the case of quasistatic fading and show how the key property in Lemma 2 can also be used to reduce the worst-case ML decoding complexity. We start by defining the intermediate variables $\mathbf{c} = [x_1, x_4]^\top$, $\mathbf{d} = [x_2, x_3]^\top$, $\mathbf{z}_{12} = [z_1, z_2]^\top$ and $\mathbf{z}_{34} = [z_3, z_4]^\top$, where $\mathbf{z} = \mathbf{Q}^*\mathbf{y}$. The ML decision minimizes the cost function

$$\begin{aligned} P(\mathbf{x}) &= \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \\ &= \|\mathbf{z} - \mathbf{R}\mathbf{x}\|^2 \\ &= \|\mathbf{z}_{12} - \mathbf{A}\mathbf{c} - \mathbf{B}\mathbf{d}\|^2 + \|\mathbf{z}_{34} - \mathbf{D}\mathbf{d}\|^2. \end{aligned} \tag{5.24}$$

The last equality follows from Lemma 5.2. Therefore, the ML decisions $\hat{\mathbf{c}}$ and $\hat{\mathbf{d}}$ can also be found recursively using:

$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}\in\mathcal{A}^2} \{\|\mathbf{z}_{12} - \mathbf{A}\mathbf{c}_*(\mathbf{d}) - \mathbf{B}\mathbf{d}\|^2 + \|\mathbf{z}_{34} - \mathbf{D}\mathbf{d}\|^2\}, \tag{5.25}$$

$$\hat{\mathbf{c}} = \mathbf{c}_*(\hat{\mathbf{d}}), \tag{5.26}$$

where

$$\mathbf{c}_*(\mathbf{d}) = \arg\min_{\mathbf{c}\in\mathcal{A}^2} \{\|\mathbf{z}_{12} - \mathbf{A}\mathbf{c} - \mathbf{B}\mathbf{d}\|^2\}. \tag{5.27}$$

Similar to $\mathbf{a}_*(\mathbf{b})$ in (4.10), the function $\mathbf{c}_*(\mathbf{d})$ in (5.27) can be viewed as producing the best $\mathbf{c}$ for a given $\mathbf{d}$. With this interpretation, the optimization in (5.25) can be viewed as that of finding the best $\mathbf{d}$ when $\mathbf{c}$ is optimized.

The optimization (5.27) is equivalent to ML detection for a channel $\mathbf{A}$ with an input of $\mathbf{c}$ and an output:

$$\mathbf{v} = \mathbf{z}_{12} - \mathbf{B}\mathbf{d}. \tag{5.28}$$

Because the asymmetric golden code induces the special property that $\mathbf{A}$ is diagonal and real, we can determine the four components of $\mathbf{c}$ *independently* from each other. Specifically, we may rewrite (5.27) as:

$$\begin{aligned}
\mathbf{c}_*(\mathbf{d}) &= \arg\min_{\mathbf{c}\in\mathcal{A}^2}\{\|\mathbf{v}-\mathbf{A}\mathbf{c}\|^2\} \\
&= \arg\min_{x_4^R\in(\mathcal{A}^R)}\{|v_2^R - r_{2,2}x_4^R|^2\} + i\cdot\arg\min_{x_4^I\in(\mathcal{A}^I)}\{|v_2^I - r_{2,2}x_4^I|^2\} + \\
&\quad \arg\min_{x_1^R\in(\mathcal{A}^R)}\{|v_1^R - r_{1,1}x_1^R|^2\} + i\cdot\arg\min_{x_1^I\in(\mathcal{A}^I)}\{|v_1^I - r_{1,1}x_1^I|^2\}.
\end{aligned} \tag{5.29}$$

Thus, the optimization in (5.27) decomposes into four independent optimizations of (5.29). Each optimization in (5.29) can be implemented with a slicer. Hence, the overall complexity of (5.29) is $\mathcal{O}(1)$. We thus arrive at the fast decodability theorem on quasistatic channels.

Theorem 5.3. *(The Asymmetric Golden Code is Fast-Decodable on Quasistatic Fading): A maximum-likelihood decoder for the asymmetric golden code with q-ary QAM alphabet can be implemented with a worst-case complexity of $\mathcal{O}(q^2)$.*

Proof. As described in (5.25), the ML decision can be found by stepping through each of the q^2 candidate values for $\mathbf{d}$, and for each implement the $\mathcal{O}(1)$ optimization of (5.29). □

5.3 Fast ML Decoding with Low Average Decoding Complexity

The decoding strategy used to prove the fast-decodable theorems has a low worst-case complexity but a high average complexity. In this section we present an efficient implementation of an ML decoder for the asymmetric golden code in quasistatic fading. The proposed ML decoder has a low average complexity and a worst-case complexity of $\mathcal{O}(q^2)$. An efficient ML decoder of the golden code for time-varying channel was proposed in Chapter 4, with a worst-case complexity of $\mathcal{O}(q^{2.5})$. Importantly, the algorithm in Figure 4.2 can be used to decode the asymmetric golden code in time-varying fading without any modification to the algorithm. The inputs to the decoding algorithm, which are the received vector and

the effective channel matrix are given by (5.16) and (5.17), respectively. Therefore, we will only consider efficient ML decoding in quasistatic fading.

A conventional sphere decoder applied to the asymmetric golden code results in a four-level tree, with a different $x_i, i \in \{1,2,3,4\}$, associated with each level. We propose a *three-level* tree that associates x_3 with the first level, x_2 with the second level and $\mathbf{c} = [x_1, x_4]^\top$ with the third level, as illustrated in Figure 5.1. This new tree is a direct result of the fact that $\mathbf{A}$ and $\mathbf{D}$ are real and diagonal (Lemma 5.2), which allows us to rewrite the ML cost function from (5.24) as

$$\begin{aligned} P(\mathbf{x}) &= \overbrace{\underbrace{\|v_1^R - r_{1,1}x_1^R\|^2}_{P_1^R} + \underbrace{\|v_1^I - r_{1,1}x_1^I\|^2}_{P_1^I} + \underbrace{\|v_2^R - r_{2,2}x_4^R\|^2}_{P_2^R} + \underbrace{\|v_2^I - r_{2,2}x_4^I\|^2}_{P_2^I}}^{P_{1,2}} + \\ &\quad \underbrace{\|z_3 - r_{3,3}x_2\|^2}_{P_3} + \underbrace{\|z_4 - r_{4,4}x_3\|^2}_{P_4}. \end{aligned} \tag{5.30}$$

Thus, as illustrated in Figure 5.1, (5.30) shows that the total cost of a leaf node $\mathbf{x}$ decomposes into the sum of three branch metrics, where P_4, P_3 and $P_{1,2}$ denotes the branch metrics for a branch at the first, second and third stage of the tree, respectively.

The fact that $r_{3,4} = 0$ leads to a significant reduction in the complexity of the Schnorr-Euchner sorting for the first two stages of the tree. Specifically, the fact that $\mathbf{D}$ is diagonal implies that there is no interference or dependence between the symbols x_2 and x_3. Therefore, we can perform a single sort for the symbol x_3 emanating from the root, and *simultaneously* a single sort for the symbol x_2 emanating from its children.

The pseudocode of an efficient implementation of the proposed asymmetric golden code ML detector is shown in Figure 5.2. The first five lines represent initializations. In particular, the first two lines are a QR decomposition of the permuted effective channel matrix in (5.21) and the computation of $\mathbf{z}$. The squared sphere radius $\hat{P}$, which represents the smallest cost (5.24) encountered so far, is initialized to infinity to ensure ML decoding (line 3). Sorting or Schnorr-Euchner enumeration is used for faster convergence. Only two sorting operations (line 4 and line 5) are required. In the pseudocode, the complex QAM alphabet $\mathcal{A}$ is represented by an ordered list, so that $\mathcal{A}(k)$ indexes the k-th symbol in the list. We next describe the remainder of algorithm, which can be interpreted as a two-level complex

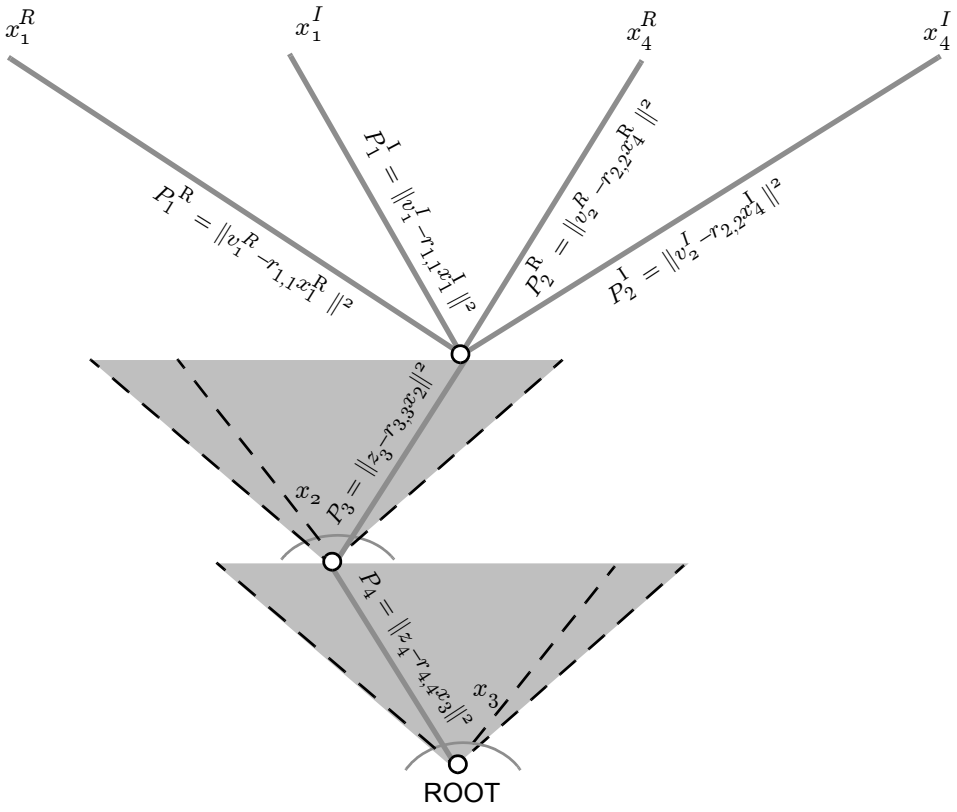


Figure 5.1: The structure of the proposed detection tree and its branch metrics for the asymmetric golden code. The cost function for the leaf node is the sum of the branch metrics, $P(\mathbf{x}) = P_1^R + P_1^I + P_2^R + P_2^I + P_3 + P_4$.

sphere decoder to choose the symbol pair $\mathbf{d} = [x_2, x_3]^\top$, followed by four independent slicers that separately decode x_1^R, x_4^R, x_1^I and x_4^I.

```
1  [Q, R] = QR decomposition(H_Π)
2  z = Q*y
3  P̂ = ∞
4  [P_4, Π_4] = sort_{a∈A}(|z_4 − r_{4,4}a|^2)
5  [P_3, Π_3] = sort_{a∈A}(|z_3 − r_{3,3}a|^2)
6  for k from 1 to q do
7      if (P_4(k) + P_3(1)) > P̂ then
8          break
9      end
10     for l from 1 to q do
11         if (P_3(l) + P_4(k)) > P̂ then
12             break
13         end
14         x_3 = A(Π_4(k))
15         x_2 = A(Π_3(l))
16         v_1 = z_1 − r_{1,3}x_2 − r_{1,4}x_3
17         v_2 = z_2 − r_{2,3}x_2 − r_{2,4}x_2
18         x_1^R = Q(v_1^R / r_{1,1})
19         x_1^I = Q(v_1^I / r_{1,1})
20         x_4^R = Q(v_2^R / r_{2,2})
21         x_4^I = Q(v_2^I / r_{2,2})
22         P_1^R = |v_1^R − r_{1,1}x_1^R|^2
23         P_1^I = |v_1^I − r_{1,1}x_1^I|^2
24         P_2^R = |v_2^R − r_{2,2}x_4^R|^2
25         P_2^I = |v_2^I − r_{2,2}x_4^I|^2
26         P = P_1^R + P_1^I + P_2^R + P_2^I + P_3(l) + P_4(k)
27         if P < P̂ then
28             P̂ = P
29             x̂ = [x_1, x_2, x_3, x_4]
30         end
31     end
32 end
```

Figure 5.2: Pseudocode of a fast ML decoder for the asymmetric golden code in quasistatic fading.

The two-level complex sphere decoder incorporates two common optimizations: radius update (line 28) and pruning (line 7, line 11). While these optimizations do not affect the worst-case complexity, they affect the average complexity significantly. The first level of the complex sphere decoder considers candidate symbols x_3 in ascending order of their branch

metric P_4 (line 6). The second level of the complex sphere decoder considers candidate symbols x_2 in ascending order of their branch metric P_3 (line 10). The decoder then removes the interference caused by $\mathbf{d} = [x_2, x_3]^\top$ and forms the two intermediate variables v_1 and v_2 of (5.28), which are functions of the symbols x_1 and x_4 only (line 16 and line 17).

Following the two-level complex sphere decoder and interference cancelation, the decoder decides on the remaining four PAM symbols x_1^R, x_4^R, x_1^I and x_4^I separately using independent slicers (line 18 through line 21). The associated branch metrics are calculated in line 22 through line 25. The overall cost P for the current candidate symbol vector is updated in line 26. Radius update (line 28) and best candidate vector update (line 29) occur if the current cost P is less than the previous smallest cost $\hat{P}$ (line 27).

5.4 Numerical Results

In this section, we compare the bit-error rate (BER) performance of the asymmetric golden code with the golden code and overlaid Alamouti codes of [16][17]. To avoid clutter we do not include the overlaid Alamouti code of [18] in our comparison; we simply note that it performs slightly worse (by about 0.1 dB) than the overlaid Alamouti codes of [16][17].

In Figure 5.3, we show simulation results for quasistatic fading with ML detection implemented using the sphere decoder algorithm, assuming 4-QAM. The asymmetric golden code, which has the lowest decoding complexity under any channel condition, suffers a performance loss of 1.0 dB compared to the golden code. This is due to a reduction in the coding gain. The coding gain of the golden code is $1/\sqrt{5}$, while the coding gain of the asymmetric golden code is $1/(2\sqrt{5})$. The overlaid Alamouti codes of [16][17] also suffer a slight performance loss of 0.3 dB, which may also be attributed to a reduced coding gain given by $1/\sqrt{7}$.

In Figure 5.4, we illustrate the impact of a time-varying channel on the performance and complexity of the overlaid Alamouti and asymmetric golden code, assuming 64-QAM. The time-selectivity is quantified by $f_D T$, the maximum Doppler frequency normalized by the symbol period. The number adjacent to each point in the figure represents the corresponding value of $f_D T$. The case of quasistatic fading corresponds to $f_D T = 0$. For

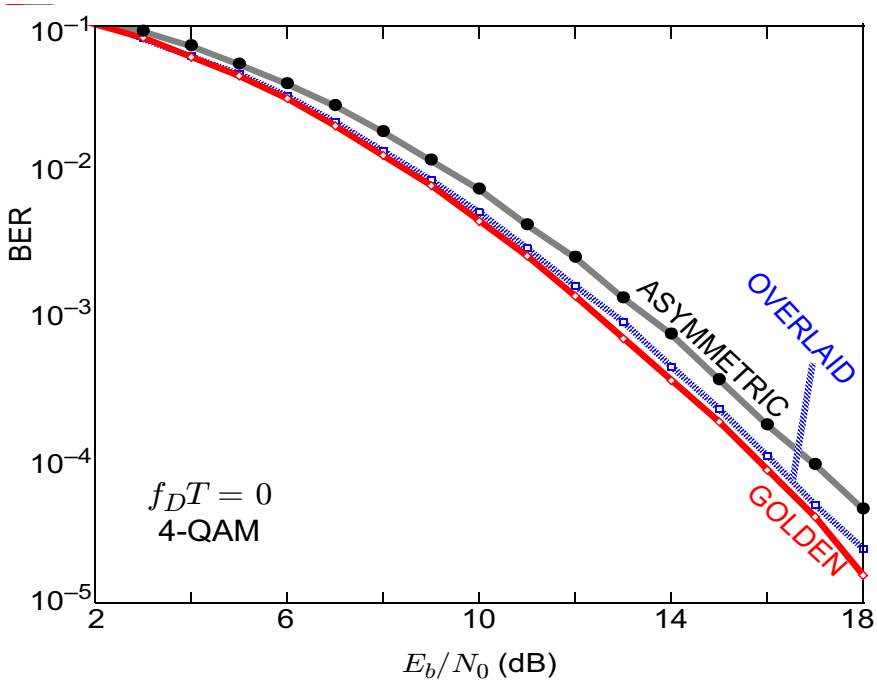


Figure 5.3: Performance comparison for the two-input two-output channel.

each space-time code, we plot a trajectory of "performance" versus "complexity" as the Doppler increases, where performance is quantified by the average SNR required to achieve BER = 10^{-3}, and where complexity is quantified by the average number of nodes visited by a complex sphere decoder that uses Schnorr-Euchner enumeration, sphere radius update and depth-first tree search. There are three trajectories:

- The left-most trajectory (labeled "overlaid + ignore") shows how the performance of the overlaid Alamouti codes of [16][17] degrades as the Doppler increases, when the receiver ignores the time variation and pretends that the channel is quasistatic. Such a strategy is ML for the special case of zero Doppler, and it is effective for slowly varying channels, but it quickly degrades as the Doppler frequency grows.

- The right-most trajectory (labeled "overlaid + ML") shows the performance and complexity of the overlaid Alamouti codes of [16][17] with an ML detector that accounts for the time variations of the channel; the performance is good for all Doppler values but the complexity is high.

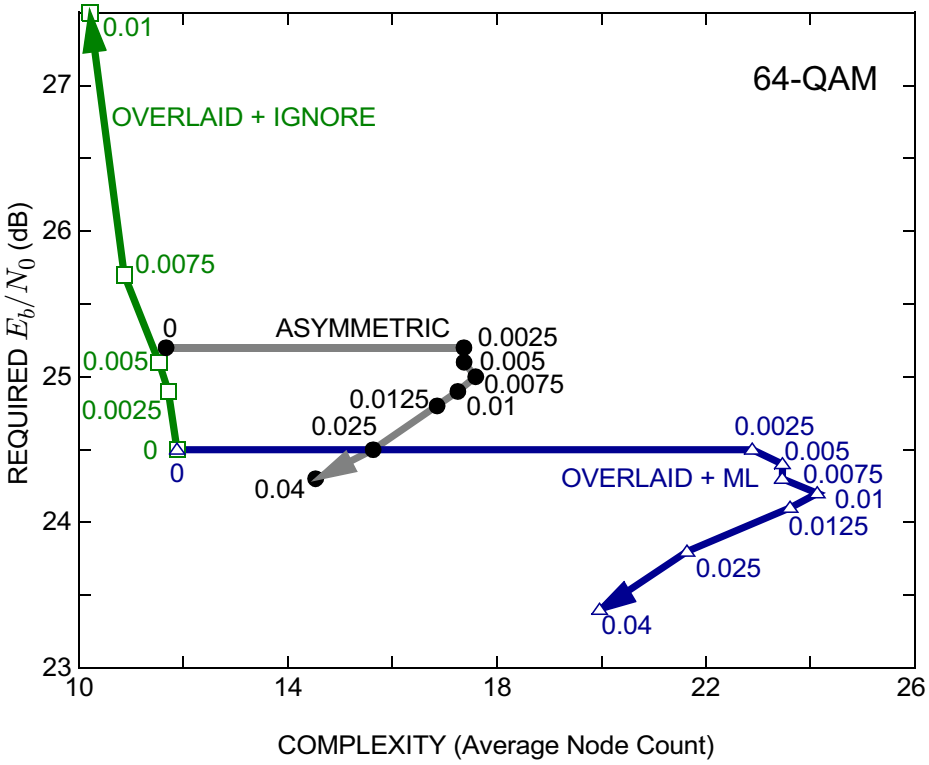


Figure 5.4: Complexity and performance as a function of normalized Doppler $f_D T$.

- The remaining trajectory (labeled "asymmetric") shows the performance and complexity of the asymmetric golden code with ML decoding. This trajectory is the only one of the three that starts in the desirable lower-left corner of the performance-complexity plane, and stays there for all values of the normalized Doppler frequency.

5.5 Conclusions

We introduced the asymmetric golden code, which asymmetrically scales the two layers of the golden code before adding. Furthermore, we proposed a fast ML decoding algorithm for the asymmetric golden code based on a unique tree construction that exploits the special structure of the effective channel matrix. The asymmetric golden code not only achieves the diversity-multiplexing tradeoff, but it maintains a low decoding complexity in both quasistatic fading and rapid time-varying fading channels.

CHAPTER 6

THE EMBEDDED ORTHOGONAL SPACE-TIME BLOCK CODES

In Chapters 4 and 5, we were concerned with the design and ML decoding of high-rate space-time block codes for the two-input two-output channel. An interesting question to consider is the choice of a space-time block code for a given rate and a given number of transmit and receive antennas. Although space-time codes that achieve arbitrary rates ranging from one to maximal rate have been proposed in the literature, these codes suffer from high decoding complexity and worse bit-error-rate performance compared to other space-time codes that have been proposed for a particular rate and a particular number of transmit and receive antennas.

In this chapter, we address the problem of designing high-rate space-time block codes for any number of transmit antennas. Specifically, we introduce the concept of *embedding*; a new concept for the construction of space-time block codes whereby information symbols of a traditional space-time block code are replaced by complex orthogonal codewords. In other words, the role of complex information symbols in the encoding process is subsumed by complex orthogonal designs.

Based on the embedding concept, we propose a family of high-rate space-time block codes called *embedded orthogonal space-time* (EOS) block codes. This family is parameterized by the number of transmit antennas, which can be any positive integer, and by the rate, which can be as high as half the number of transmit antennas. One consequence of embedding orthogonal designs is the induced orthogonality in the effective channel matrix, which leads to reduced-complexity decoding. The embedded orthogonal space-time codes have lower decoding complexity than previously reported space-time codes for any number of transmit antennas, and for any rate. Furthermore, for rates larger than one, simulation results for up to six transmit antennas show that the embedded-orthogonal space-time codes outperform all previous constructions, when performance is measured by error probability

in quasistatic Rayleigh fading.

The remainder of this chapter is organized as follows. In Section 6.1, we present the proposed family of embedded orthogonal space-time codes. In Section 6.2, we discuss a low-complexity ML decoder for the proposed codes. In Section 6.3, we present numerical results. In Section 6.4, we conclude the chapter.

6.1 The Embedded Orthogonal Space-Time Block Codes

In this section we describe the proposed embedded orthogonal space-time codes and compare to the threaded algebraic space-time codes discussed in Section 2.4.6. The design of the proposed embedded orthogonal space-time block codes is a three-step process, based on the choice of three parameters:

- a complex orthogonal design $\boldsymbol{\mathcal{G}}$ called the *embedded code*. The embedded code of size $T_1 \times M_1$, is characterized by its rate $\mathcal{R}_1$ and number of antennas M_1;
- an algebraic rotation matrix $\mathbf{G}$;
- a complex number ϕ.

Before we discuss the construction of the embedded orthogonal codes, we discuss the choice of these parameters. First, we choose orthogonal designs for embedding because they induce orthogonality between the columns of the effective channel matrix, resulting in reduced complexity decoding. Second, we choose real algebraic rotation matrices because they maximize the coding gain of an encoded thread. Furthermore, real generator matrices induce orthogonality between the real and imaginary components of the columns of the effective channel matrix, resulting in further reduction in decoding complexity. Finally, we choose ϕ to ensure a nonvanishing determinant, ensure full diversity or ensure a good bit-error rate performance.

We will present the embedded orthogonal space-time codes by drawing similarities to the perfect space-time code or threaded algebraic space-time code construction of (2.31).

Instead of defining $\mathbf{x}_\ell$ as the *vector* of M information symbols for the ℓ-th thread, we define:

$$\mathbf{X}_\ell = \begin{bmatrix} \boldsymbol{\mathcal{G}}_{\ell,1} \\ \boldsymbol{\mathcal{G}}_{\ell,2} \\ \vdots \\ \boldsymbol{\mathcal{G}}_{\ell,M_2-1} \\ \boldsymbol{\mathcal{G}}_{\ell,M_2} \end{bmatrix} \tag{6.1}$$

as the $T_1 M_2 \times M_1$ *matrix* of M_2 embedded codewords for the ℓ-th thread, where $\boldsymbol{\mathcal{G}}_{\ell,m}$, $m \in \{1, 2, \cdots, M_2\}$, is the m-th embedded orthogonal codeword of the ℓ-th thread. By comparing $\mathbf{x}_\ell$ and $\mathbf{X}_\ell$, we see that the information symbols $x_{\ell,m}$ have been replaced by orthogonal codewords $\boldsymbol{\mathcal{G}}_{\ell,m}$.

The proposed rate-$\mathcal{R}$ embedded orthogonal space-time code of size $T \times M$, where $T = T_1 M_2$ and $M = M_1 M_2$, is:

$$\mathbf{C}_{EOS} = \sum_{\ell=1}^{\mathcal{R}_2} \text{blkdiag}\left(\mathbf{U}_\ell\right) \left(\mathbf{J}^{\ell-1} \otimes \mathbf{I}_{M_1}\right), \tag{6.2}$$

where

- $\mathcal{R}_2 = \lceil \mathcal{R}/\mathcal{R}_1 \rceil$
- $\otimes$ is the Kronecker product
- blkdiag($\mathbf{U}_\ell$) is the $T_1 M_2 \times M_1 M_2$ *block* diagonal matrix with the $T_1 \times M_1$ subblocks of $\mathbf{U}_\ell$ on the diagonal
- $\mathbf{U}_\ell = \left(\mathbf{G} \otimes \mathbf{I}_{T_1}\right) \mathbf{X}_\ell = \begin{bmatrix} \boldsymbol{\mathcal{U}}_{\ell,1} \\ \boldsymbol{\mathcal{U}}_{\ell,2} \\ \vdots \\ \boldsymbol{\mathcal{U}}_{\ell,M_2-1} \\ \boldsymbol{\mathcal{U}}_{\ell,M_2} \end{bmatrix}$

- $\mathbf{G}$ is an $M_2 \times M_2$ unitary real *rotation* or *generator* matrix given by

$$\mathbf{G} = \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,M_2} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,M_2} \\ \vdots & \vdots & \ddots & \vdots \\ g_{M_2,1} & g_{M_2,2} & \cdots & g_{M_2,M_2} \end{bmatrix}. \tag{6.3}$$

- $\mathbf{J} = [\phi \mathbf{e}_{M_2}, \mathbf{e}_1, \cdots, \mathbf{e}_{M_2-1}]$

- ϕ is a unit-magnitude complex number, the value of which depends on the number of transmit antennas M.

The embedded orthogonal space-time block code for a prime number of antennas, where M cannot be factored as $M = M_1 M_2$ is obtained by deleting the last column of the embedded orthogonal code for $M+1$ antennas. For example, the embedded orthogonal space-time code for three, five and seven antennas can be obtained by deleting the last column of the embedded orthogonal space-time block code for four, six and eight transmit antennas, respectively.

The orthogonality embedding concept is evident by comparing (2.31) to (6.2). In particular, the transmitted symbols $x_{\ell,m}$ in (2.31) are replaced with orthogonal block codes $\boldsymbol{\mathcal{G}}_{\ell,m}$ in (6.2). Alternatively, the orthogonal codewords are *embedded* into the code in (2.31) to yield the embedded orthogonal codeword of (6.2).

Having discussed the concept of embedding and why we embed orthogonal designs, we next discuss the choice of the three parameters in the design of embedded orthogonal space-time block codes, namely the embedded code $\boldsymbol{\mathcal{G}}$, the rotation matrix $\mathbf{G}$ and the complex number ϕ. We start with the specific choice of the embedded complex orthogonal design.

The maximal rate of complex orthogonal designs for arbitrary number of antennas and a systematic construction of these maximal rate codes is given in [26]. Of particular importance to us are the rate-1 Alamouti code for two antennas and rate-3/4 codes for three and four antennas given by (2.20) in Section 2.4.1. Although the embedded orthogonal space-time encoder in (6.2) allows for arbitrary orthogonal codes for any number of antennas, we will only consider the embedding of orthogonal designs for $M_1 \in \{2, 3, 4\}$. This is because,

as discussed in Section 2.4.1 and shown in Table 2.1, the orthogonal code rate tends to $\frac{1}{2}$ as the number of antennas increases, and the code length becomes prohibitively large for practical implementations. As a result, the rate of the embedded orthogonal space-time code decreases, while the decoding latency and complexity increase when we embed orthogonal designs for more than four antennas.

Regarding the choice of $\mathbf{G}$, we will use the algebraic rotation matrices in [56][57]. These algebraic rotations matrices are the best known rotations in terms of maximizing the coding gain of a single encoded layer $\texttt{diag}(\mathbf{Gx})$. Consequently, DAST, TAST and perfect codes use them to encode the different threads in (2.23), (2.31) and (2.32), respectively. We will also use the rotation matrices in [56][57] to construct the embedded orthogonal space-time block code.

Regarding the choice of ϕ, we used partial search over a random subset of all possible codewords to find the value of ϕ that maximizes the coding gain in 2.12. We used partial search because we could not optimizing the value of ϕ analytically, and even numerical exhaustive-search was computationally infeasible for large number of antennas and/or large alphabet sizes. We observed that we could borrow the unit-magnitude complex number ϕ from the full-rate TAST code, except for $M \in \{4, 8\}$, where we will use $\phi = e^{i\pi/2}$, and obtain an embedded orthogonal code with high coding gain. We note that our choice of ϕ does not guarantee full diversity. Simulations results for up to six transmit antennas, however, show that the embedded orthogonal code outperforms the perfect and TAST codes in terms of the SNR required to achieve a target error probability.

An interesting question to consider is the maximum rate of the embedded orthogonal design. We answer this question in the following lemma.

Lemma 6.1. *The maximum achievable rate of the embedded orthogonal space-time code in (6.2) for $M = M_1 M_2$ antennas is $\mathcal{R}_{max} = \frac{\mathcal{R}_1}{M_1} \times M = \frac{M}{2}$.*

Proof. In general, the maximum achievable rate for a space-time code with M antennas is simply M since we can transmit one symbol, from each antenna, in each signaling interval. For the embedded orthogonal code, however, we can effectively only transmit $\frac{\mathcal{R}_1}{M_1}$ symbols,

per signaling interval, per antenna. Therefore, the maximum achievable rate for M antennas is

$$\mathcal{R}_{max} = \frac{\mathcal{R}_1}{M_1} \times M. \tag{6.4}$$

For orthogonal designs, however, we have $\mathcal{R}_1 \leq 1$ and $M_1 \geq 2$. Therefore, $\frac{\mathcal{R}_1}{M_1} \leq \frac{1}{2}$. Substituting for the upper bound on $\frac{\mathcal{R}_1}{M_1}$ in (6.4), we arrive at the desired result that $\mathcal{R}_{max} = \frac{M}{2}$. □

Corollary 6.1. *When the embedded code is the Alamouti code, the rate-$\frac{3}{4}$ orthogonal code for three antennas or the rate-$\frac{3}{4}$ orthogonal code for four antennas, then the maximum achievable rate of the embedded orthogonal space-time block code in (6.2) for $M = M_1 M_2$ antennas is $\mathcal{R}_{max} = \frac{1}{2}M$, $\mathcal{R}_{max} = \frac{1}{4}M$, and $\mathcal{R}_{max} = \frac{3}{16}M$, respectively.*

We make several important remarks regarding the construction of the embedded orthogonal space-time block codes.

- If the ratio $\mathcal{R}/\mathcal{R}_1$ is not an integer, then the rate of the embedded orthogonal space-time code in (6.2) is higher than $\mathcal{R}$. A rate-$\mathcal{R}$ embedded orthogonal code can then be obtained by puncturing the embedded codewords in the $\mathcal{R}_2$-th thread. We will see an example of a punctured fractional rate embedded orthogonal code later in the section.

- When the parameter $M_2 = 1$, then we have

$$\mathbf{U}_\ell = \mathbf{X}_\ell = \boldsymbol{\mathcal{G}}_{\ell,1},\ \mathbf{G} = [\,1\,],\ \text{and}\ \mathbf{J} = [\,1\,]. \tag{6.5}$$

  Therefore, there is no embedding, and the embedded orthogonal space-time code reduces to the orthogonal design.

- When the parameter $\mathcal{R}_2 = 1$, and the embedded code is the Alamouti space-time code, then the embedded orthogonal space-time code simplifies to the semi-orthogonal algebraic space-time (SAST) code given in (2.30). Specifically, the embedded orthogonal space-time code in (6.2) simplifies to the SAST code in (2.30) to within a unitary transformation as follows

$$\mathbf{C}_{EOS} = \mathbf{\Pi}_L \mathbf{C}_{SAST} \mathbf{\Pi}_R, \tag{6.6}$$

where $\mathbf{\Pi}_R = [\mathbf{e}_1, \mathbf{e}_{1+\frac{M}{2}}, \mathbf{e}_2, \mathbf{e}_{2+\frac{M}{2}}, \ldots, \mathbf{e}_{\frac{M}{2}}, \mathbf{e}_M]$ and $\mathbf{\Pi}_L = \mathbf{\Pi}_R^\top$. When $\mathcal{R}_2 = 1$ and the embedded code is the Alamouti code with $\mathcal{R}_1 = 1$, we have that $\mathcal{R} \leq 1$. Therefore, the proposed embedded orthogonal space-time code has the same diversity, coding gain, rate, and decoding complexity as the SAST code when $\mathcal{R} = 1$ and the embedded code is the Alamouti code, albeit with a larger peak-to-average power ratio. Therefore, the embedded orthogonal space-time code subsumes the SAST code as a special case.

- The embedded orthogonal space-time block is equivalent to the single-symbol decodable code of [58] for $M = 4$, $\mathcal{R} = 1$ and the Alamouti embedded code. This follows immediately from the fact that the single-symbol decodable code of [58] is equivalent to SAST code for four transmit antennas.

- Orthogonality embedding offers another advantage in addition to inducing orthogonality in the effective channel matrix. Specifically, when compared to the perfect space-time and TAST codes, the embedded orthogonal codes have a few number of zeros in the space-time code matrix because we encode the element and its conjugate. Hence, we have a lower peak-to-average power ratio. Furthermore, for the same number of antennas M, the unitary generator matrix $\mathbf{G}$ of the embedded orthogonal code has size $M_2 \times M_2$. For the perfect and TAST codes, the generator matrix $\mathbf{G}$ has size $M \times M$. Since $M_1 \geq 2$ and $M = M_1 M_2$, we have that $M_2 < M$. Therefore, the embedded orthogonal code linearly combines a fewer number of symbols compared to the perfect or TAST codes, resulting in even a further reduction in peak-to-average power ratio.

For clarity of exposition, we next discuss three examples for the construction of embedded orthogonal space-time codes. Specifically, we discuss the encoding of rate-2, rate-3, and rate-$\frac{3}{2}$ embedded orthogonal codes for four, five and eight transmit antennas, respectively.

Example 6.1. We construct the rate-2 embedded orthogonal space-time code for four transmit antennas. Since $M = M_1 M_2 = 4$, there is only one choice for the orthogonal code, which is the Alamouti code with $M_1 = 2$. This is because M_1 cannot be 3 since M_2 has to

be an integer; furthermore, M_1 cannot be 4 because it implies that $M_2 = 1$ such that there is no embedding, and we simply obtain the rate-3/4 orthogonal design.

With $M_1 = 2$ and $\mathcal{R}_1 = 1$, we have that $M_2 = 2$ and $\mathcal{R}_2 = 2$, and hence, $\mathbf{G}$ is the 2×2 real generator matrix:

$$\mathbf{G} = \begin{bmatrix} \alpha_1 & \alpha_2 \\ -\alpha_2 & \alpha_1 \end{bmatrix} = \begin{bmatrix} 0.851 & 0.526 \\ -0.526 & 0.851 \end{bmatrix}, \tag{6.7}$$

where $\alpha_1 = \cos(\theta)$, $\alpha_2 = \sin(\theta)$ and $\theta = \frac{1}{2}\tan^{-1}(2)$. The generator matrix in (6.7) maximizes the coding gain compared to all real unitary generator matrices. The generator matrix in (6.7) is the same as the generator matrix $\mathbf{G}$ for the golden code, asymmetric golden code and the diagonal algebraic space-time code (see (4.2) and (5.3)). The rate-2 embedded orthogonal space-time code for four transmit antennas is then given by

$$\mathbf{C} = \begin{bmatrix} \boldsymbol{\mathcal{U}}_{1,1} & \boldsymbol{\mathcal{U}}_{2,1} \\ \phi\boldsymbol{\mathcal{U}}_{2,2} & \boldsymbol{\mathcal{U}}_{1,2} \end{bmatrix} = \begin{bmatrix} \alpha_1\boldsymbol{\mathcal{G}}_{1,1} + \alpha_2\boldsymbol{\mathcal{G}}_{1,2} & \alpha_1\boldsymbol{\mathcal{G}}_{2,1} + \alpha_2\boldsymbol{\mathcal{G}}_{2,2} \\ \phi\left(-\alpha_2\boldsymbol{\mathcal{G}}_{2,1} + \alpha_1\boldsymbol{\mathcal{G}}_{2,2}\right) & -\alpha_2\boldsymbol{\mathcal{G}}_{1,1} + \alpha_1\boldsymbol{\mathcal{G}}_{1,2} \end{bmatrix}, \tag{6.8}$$

where $\phi = e^{i\pi/2}$ and $\boldsymbol{\mathcal{G}}_{\ell,m}$ is the m-th Alamouti space-time block codeword, $m \in \{1, 2\}$, for the ℓ-th layer, $\ell \in \{1, 2\}$. By comparison, the rate-2 TAST or perfect code for two transmit antennas is

$$\mathbf{C} = \begin{bmatrix} u_{1,1} & u_{2,1} \\ \phi u_{2,2} & u_{1,2} \end{bmatrix} = \begin{bmatrix} \alpha_1 x_{1,1} + \alpha_2 x_{1,2} & \alpha_1 x_{2,1} + \alpha_2 x_{2,2} \\ \phi\left(-\alpha_2 x_{2,1} + \alpha_1 x_{2,2}\right) & -\alpha_2 x_{1,1} + \alpha_1 x_{1,2} \end{bmatrix}, \tag{6.9}$$

where $\phi = e^{i\pi/2}$ for the perfect space-time code and $\phi = e^{i\pi/6}$ for the TAST code. By comparing (6.8) and (6.9), we see that the information symbols $x_{\ell,m}$ of the perfect code in (6.9) have been replaced with Alamouti blocks $\boldsymbol{\mathcal{G}}_{\ell,m}$ in (6.8). Each $\boldsymbol{\mathcal{G}}_{\ell,m}$ is a 2×2 matrix containing two complex information symbols, so the matrix of (6.8) is 4×4 and encodes a total of 8 symbols. □

We next discuss the encoding of the rate-3 embedded orthogonal space-time code for five transmit antennas. The example illustrates the construction of embedded orthogonal codes when the number of antennas M is prime.

Example 6.2. We construct the rate-3 embedded orthogonal space-time code for five transmit antennas. We first construct the embedded orthogonal code for $5+1=6$ antennas. The embedded orthogonal code for five antennas is then obtained by deleting the last column of the embedded orthogonal code for $M=6$ antennas.

Since $M=M_1M_2=6$, there is only one choice for the orthogonal code; the Alamouti code with $M_1=2$. Although $M_1=3$ is a potential choice, it implies that the maximum rate is $\mathcal{R}=\frac{1}{4}\times 6=1.5$, which is less than the desired rate $\mathcal{R}=3$ (see Corollary 6.1).

With $M_1=2$ and $\mathcal{R}_1=1$, we have that $M_2=3$ and $\mathcal{R}_2=3$, and hence, $\mathbf{G}$ is the 3×3 real generator matrix:

$$\mathbf{G}=\begin{bmatrix} -0.328 & -0.591 & -0.737 \\ -0.737 & -0.328 & 0.591 \\ -0.591 & 0.737 & -0.328 \end{bmatrix}. \tag{6.10}$$

The generator matrix in (6.10) is the same as the generator matrix $\mathbf{G}$ for the threaded algebraic and diagonal algebraic space-time codes. The rate-3 embedded orthogonal space-time code for six transmit antennas is then given by

$$\mathbf{C}=\begin{bmatrix} \boldsymbol{u}_{1,1} & \boldsymbol{u}_{2,1} & \boldsymbol{u}_{3,1} \\ \phi\boldsymbol{u}_{3,2} & \boldsymbol{u}_{1,2} & \boldsymbol{u}_{2,2} \\ \phi\boldsymbol{u}_{2,3} & \phi\boldsymbol{u}_{3,3} & \boldsymbol{u}_{1,3} \end{bmatrix}. \tag{6.11}$$

The embedded orthogonal space-time block code for five antennas is obtained by deleting the last column of the code matrix in (6.11). □

We next discuss an example of a rate-$\frac{3}{2}$ embedded orthogonal code for eight transmit antennas, wherein we can embed either the Alamouti code or the rate-$\frac{3}{4}$ orthogonal code for four transmit antennas. We also illustrate how the Alamouti embedded code can be punctured to achieve the rate-$\frac{3}{2}$ embedded orthogonal space-time code.

Example 6.3. We construct the rate-$\frac{3}{2}$ embedded orthogonal space-time code for eight transmit antennas. For $M=M_1M_2=8$, there are two choices for the embedded orthogonal code; the rate-1 Alamouti code with $M_1=2$, or the rate-$\frac{3}{4}$ orthogonal code with $M_1=4$. We next consider these two choices for the embedded code.

Case 1: We first consider the rate-$\frac{3}{4}$ orthogonal code as the embedded code. This implies that $M_2 = 2$ and $\mathcal{R}_2 = 2$, as was the case in Example 6.1. Therefore, the rate-$\frac{3}{2}$ embedded orthogonal space-time code for eight transmit antennas is

$$\mathbf{C} = \begin{bmatrix} \boldsymbol{\mathcal{U}}_{1,1} & \boldsymbol{\mathcal{U}}_{2,1} \\ \phi\boldsymbol{\mathcal{U}}_{2,2} & \boldsymbol{\mathcal{U}}_{1,2} \end{bmatrix} = \begin{bmatrix} \alpha_1\boldsymbol{\mathcal{G}}_{1,1} + \alpha_2\boldsymbol{\mathcal{G}}_{1,2} & \alpha_1\boldsymbol{\mathcal{G}}_{2,1} + \alpha_2\boldsymbol{\mathcal{G}}_{2,2} \\ \phi\left(-\alpha_2\boldsymbol{\mathcal{G}}_{2,1} + \alpha_1\boldsymbol{\mathcal{G}}_{2,2}\right) & -\alpha_2\boldsymbol{\mathcal{G}}_{1,1} + \alpha_1\boldsymbol{\mathcal{G}}_{1,2} \end{bmatrix}, \tag{6.12}$$

where α_1, α_2, and ϕ are as in Example 6.1, and $\boldsymbol{\mathcal{G}}_{\ell,m}$ is the m-th rate-$\frac{3}{4}$ orthogonal space-time block codeword for four antennas given in (2.20), $m \in \{1, 2\}$, for the ℓ-th layer, $\ell \in \{1, 2\}$. Orthogonality embedding is again evident by comparing (6.9) and (6.12). In particular, the information symbols of the perfect or TAST code in (6.9) have been replaced with the rate-$\frac{3}{4}$ orthogonal codewords in (6.9).

We note that the rate-2 golden code for two transmit antennas in (6.9), the rate-2 embedded orthogonal code for four transmit antennas in (6.8), and the rate-$\frac{3}{2}$ embedded orthogonal code for eight transmit antennas in (6.12) have the same matrix form. They only differ in the choice of the embedded code. Specifically, the golden code encodes complex information symbols $x_{\ell,m}$, the rate-2 embedded orthogonal code for four transmit antennas embeds the rate-1 Alamouti space-time code, and the rate-$\frac{3}{2}$ embedded orthogonal code for eight transmit antennas embeds the rate-$\frac{3}{4}$ orthogonal code for four transmit antennas.

Case 2: We consider the rate-1 Alamouti code as the embedded code. We have $M_2 = 4$ and $\mathcal{R}_2 = 2$, and hence, $\mathbf{G}$ is the 4×4 real generator matrix given by

$$\mathbf{G} = \begin{bmatrix} 0.405 & 0.542 & -0.656 & -0.335 \\ 0.273 & 0.498 & 0.169 & 0.806 \\ 0.335 & -0.656 & -0.542 & 0.405 \\ 0.806 & -0.169 & 0.498 & -0.273 \end{bmatrix}. \tag{6.13}$$

The rate-2 embedded orthogonal space-time code for eight transmit antennas is then given by

$$\mathbf{C} = \begin{bmatrix} \boldsymbol{\mathcal{U}}_{1,1} & \boldsymbol{\mathcal{U}}_{2,1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{U}}_{1,2} & \boldsymbol{\mathcal{U}}_{2,2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{U}}_{1,3} & \boldsymbol{\mathcal{U}}_{2,3} \\ \phi\boldsymbol{\mathcal{U}}_{2,4} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{U}}_{1,4} \end{bmatrix}, \tag{6.14}$$

where $\phi = e^{i\pi/2}$. In order to obtain the rate-$\frac{3}{2}$ embedded orthogonal code, we puncture the second embedded thread $\mathbf{U}_2$ to obtain

$$\mathbf{C} = \begin{bmatrix} \boldsymbol{\mathcal{U}}_{1,1} & \bar{\boldsymbol{\mathcal{U}}}_{2,1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{U}}_{1,2} & \bar{\boldsymbol{\mathcal{U}}}_{2,2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{U}}_{1,3} & \bar{\boldsymbol{\mathcal{U}}}_{2,3} \\ \phi\bar{\boldsymbol{\mathcal{U}}}_{2,4} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{U}}_{1,4} \end{bmatrix}, \tag{6.15}$$

where

- $\bar{\mathbf{U}}_\ell = (\mathbf{G} \otimes \mathbf{I}_{T_1})\, \bar{\mathbf{X}}_\ell = \begin{bmatrix} \bar{\boldsymbol{\mathcal{U}}}_{\ell,1} \\ \vdots \\ \bar{\boldsymbol{\mathcal{U}}}_{\ell,M_2} \end{bmatrix}$
- $\bar{\mathbf{X}}_\ell = \begin{bmatrix} \bar{\boldsymbol{\mathcal{G}}}_{\ell,1} \\ \vdots \\ \bar{\boldsymbol{\mathcal{G}}}_{\ell,M_2} \end{bmatrix}$
- $\bar{\boldsymbol{\mathcal{G}}}_{\ell,m}$ is the rate-$\frac{1}{2}$ punctured Alamouti block codeword defined by

$$\bar{\boldsymbol{\mathcal{G}}}_{\ell,m} = \begin{bmatrix} x_{\ell,m} & 0 \\ 0 & x^*_{\ell,m} \end{bmatrix}. \tag{6.16}$$

□

We summarize the three design parameters for the construction of the embedded orthogonal space-time block codes for up to eight transmit antennas in Table 6.1. In describing the embedded code, we use the notation $\boldsymbol{\mathcal{G}}(M_1, \mathcal{R}_1)$ to denote the rate-$\mathcal{R}_1$ orthogonal code designed for M_1 antennas, and we use $\bar{\boldsymbol{\mathcal{G}}}(2, \frac{1}{2})$ to denote the rate-$\frac{1}{2}$ punctured Alamouti code in (6.16).

6.2 Fast ML Decoding of the Embedded Orthogonal Space-Time Codes

In this section, we will discuss maximum-likelihood decoding of the embedded orthogonal space-time block code. We will assume that $N = \lceil \mathcal{R} \rceil$ to ensure reliable detection at the receiver. The received vector at the n-th receive antenna during time slots $t \in \{1, \cdots, T\}$

Table 6.1: Design Parameters for Embedded Orthogonal Space-Time Block Codes.

M	$\mathcal{R}$	Embedded Code	$\mathbf{G}$	ϕ
2	1	$\boldsymbol{\mathcal{G}}(2,1)$	[1]	1
4	0.75	$\boldsymbol{\mathcal{G}}(4,\frac{3}{4})$	[1]	1
	1	$\boldsymbol{\mathcal{G}}(2,1)$	(6.7)	1
	1.5	$\boldsymbol{\mathcal{G}}(2,1)$,$\bar{\boldsymbol{\mathcal{G}}}(2,\frac{1}{2})$	(6.7)	$e^{i\pi/2}$
	2	$\boldsymbol{\mathcal{G}}(2,1)$	(6.7)	$e^{i\pi/2}$
6	0.75	$\boldsymbol{\mathcal{G}}(3,\frac{3}{4})$	(6.7)	1
	1	$\boldsymbol{\mathcal{G}}(2,1)$	(6.10)	1
	1.5	$\boldsymbol{\mathcal{G}}(3,\frac{3}{4})$	(6.7)	$e^{i\pi/12}$
		$\boldsymbol{\mathcal{G}}(2,1)$, $\bar{\boldsymbol{\mathcal{G}}}(2,\frac{1}{2})$	(6.10)	$e^{i\pi/12}$
	2	$\boldsymbol{\mathcal{G}}(2,1)$	(6.10)	$e^{i\pi/12}$
	3	$\boldsymbol{\mathcal{G}}(2,1)$	(6.10)	$e^{i\pi/12}$
8	0.75	$\boldsymbol{\mathcal{G}}(4,\frac{3}{4})$	(6.7)	1
	1	$\boldsymbol{\mathcal{G}}(2,1)$	(6.13)	1
	1.5	$\boldsymbol{\mathcal{G}}(4,\frac{3}{4})$	(6.7)	$e^{i\pi/2}$
		$\boldsymbol{\mathcal{G}}(2,1)$, $\bar{\boldsymbol{\mathcal{G}}}(2,\frac{1}{2})$	(6.13)	$e^{i\pi/2}$
	2	$\boldsymbol{\mathcal{G}}(2,1)$	(6.13)	$e^{i\pi/2}$
	3	$\boldsymbol{\mathcal{G}}(2,1)$	(6.13)	$e^{i\pi/2}$
	4	$\boldsymbol{\mathcal{G}}(2,1)$	(6.13)	$e^{i\pi/2}$

can be written as

$$\mathbf{y}_n = \mathbf{C}_{EOS}\mathbf{h}_n + \mathbf{w}_n, \tag{6.17}$$

where $\mathbf{h}_n = [h_{1,n}, h_{2,n}, \cdots, h_{M,n}]^\top$, and $\mathbf{y}_n = [y_n[1], y_n[2], \cdots, y_n[T]]^\top$ and $\mathbf{w}_n = [w_n[1], w_n[2], \cdots, w_n[T]]^\top$ are the vectors of received signals and noise at the n-th receive antenna, respectively. The received signal from all antennas at all time intervals is then given by:

$$\begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} = (\mathbf{I}_N \otimes \mathbf{C}_{EOS}) \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_N \end{bmatrix} + \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_N \end{bmatrix}. \tag{6.18}$$

As mentioned in Section 2.1, the system in (6.18) should be expressed in the complex-valued form in (2.6) or the real-valued form in (2.7) in order to facilitate the use of efficient decoding algorithms. Since the representation in (6.18) is a linear system of equations in the complex information symbols *and* their conjugates, it is not always possible to express (6.18) as a function of only the complex information symbols as in (2.6). Therefore, we will use the real-valued system model in (2.7). Specifically, we have

$$\check{\mathbf{y}} = \check{\mathbf{H}}\check{\mathbf{x}} + \check{\mathbf{w}}, \tag{6.19}$$

where

- $\check{\mathbf{y}} = [y_1^R[1], y_1^I[1], \cdots, y_1^R[T], y_1^I[T], \cdots, y_N^R[1], y_N^I[1], \cdots, y_N^R[T], y_N^I[T]]^\top$ is the *real* vector of size $2NT \times 1$ of received samples at all time intervals, from all receive antennas, where $T = T_1 M_2$
- $\check{\mathbf{H}}$ is the $2NT \times 2\mathcal{R}T$ *real* effective channel matrix
- $\check{\mathbf{x}} = [x_1^R, x_1^I, x_2^R, x_2^I, \cdots, x_{\mathcal{R}T}^R, x_{\mathcal{R}T}^I]^\top$ is the $2\mathcal{R}T \times 1$ vector of transmitted *real* information symbols
- $\check{\mathbf{w}} = [w_1^R[1], w_1^I[1], \cdots, w_1^R[T], w_1^I[T], \cdots, w_N^R[1], w_N^I[1], \cdots, w_N^R[T], w_N^I[T]]^\top$ is the real vector of size $2NT \times 1$ of noise samples at all time intervals, from all receive antennas

We next discuss the general form of the effective channel matrix in (6.19) of the embedded orthogonal space-time block code.

6.2.1 The Effective Channel Matrix of the Embedded Orthogonal Space-Time Block Code

In determining the general form of the effective channel matrix of the embedded orthogonal codes, we will first introduce the notational convention. Next, we will determine the form of the effective channel matrices for the orthogonal space-time block codes used in the embedding or construction of the embedded orthogonal codes. Finally, we will determine the general form of the effective channel matrix of the embedded orthogonal space-time block code by drawing similarities to effective channel matrix of the TAST code.

We begin by introducing the notation that will be used throughout the discussion. In Section 6.1, we introduced the notation $\boldsymbol{\mathcal{G}}_{\ell,m}$ to designate the m-th embedded orthogonal codeword of the ℓ-th thread. Similarly, for the Alamouti code, the rate-$\frac{3}{4}$ orthogonal code for three antennas and the rate-$\frac{3}{4}$ orthogonal code for four antennas, we will use the notation $\boldsymbol{\mathcal{H}}_{n,m}$ to designate the m-th real-valued effective channel matrix of size $2T_1 \times 2T_1\mathcal{R}_1$ for the n-th receive antenna that is a function of the fading coefficients $h_{M_1(m-1)+1,n}$ through $h_{mM_1,n}$. We will also introduce $\mathcal{F}_R(\cdot)$ as the function that converts a complex-valued matrix into the real-valued equivalent. Specifically, for a complex matrix $\mathbf{C}$ of size $T \times M$, the function $\mathcal{F}_R(\mathbf{C})$ is defined as

$$\mathcal{F}_R(\mathbf{C}) = \begin{bmatrix} c_{1,1}^R & -c_{1,1}^I & c_{1,2}^R & -c_{1,2}^I & \cdots & c_{1,M}^R & -c_{1,M}^I \\ c_{1,1}^I & c_{1,1}^R & c_{1,2}^I & c_{1,2}^R & \cdots & c_{1,M}^I & c_{1,M}^R \\ c_{2,1}^R & -c_{2,1}^I & c_{2,2}^R & -c_{2,2}^I & \cdots & c_{2,M}^R & -c_{2,M}^I \\ c_{2,1}^I & c_{2,1}^R & c_{2,2}^I & c_{2,2}^R & \cdots & c_{2,M}^I & c_{2,M}^R \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \\ c_{T,1}^R & -c_{T,1}^I & c_{T,2}^R & -c_{T,2}^I & \cdots & c_{T,M}^R & -c_{T,M}^I \\ c_{T,1}^I & c_{T,1}^R & c_{T,2}^I & c_{T,2}^R & \cdots & c_{T,M}^I & c_{T,M}^R \end{bmatrix}, \tag{6.20}$$

where $c_{k,l}$ is the complex element in the k-th row and l-th column. We can see that every 2×2 subblock in (6.20) is the real-valued matrix representation of a complex number.

Next, we will determine the form of the real-valued effective channel matrices for the Alamouti code, the rate-$\frac{3}{4}$ orthogonal code for three antennas and rate-$\frac{3}{4}$ orthogonal code for four antennas. We will use these effective channel matrices to construct the general form of the real-valued effective channel of the embedded orthogonal space-time block code.

Given the code matrix for the Alamouti code in (2.20) and the system model in (6.19), the real-valued effective channel matrix is

$$\boldsymbol{\mathcal{H}}_{n,m} = \begin{bmatrix} h_{2m-1,n}^R & -h_{2m-1,n}^I & h_{2m,n}^R & -h_{2m,n}^I \\ h_{2m-1,n}^I & h_{2m-1,n}^R & h_{2m,n}^I & h_{2m,n}^R \\ h_{2m,n}^R & h_{2m,n}^I & -h_{2m-1,n}^R & -h_{2m-1,n}^I \\ h_{2m,n}^I & -h_{2m,n}^R & -h_{2m-1,n}^I & h_{2m-1,n}^R \end{bmatrix}. \tag{6.21}$$

The real-valued effective channel matrix for the rate-$\frac{3}{4}$ orthogonal code for three antennas is

$$\boldsymbol{\mathcal{H}}_{n,m} = \begin{bmatrix} h^R_{3m-2,n} & -h^I_{3m-2,n} & -h^R_{3m,n} & -h^I_{3m,n} & 0 & 0 \\ h^I_{3m-2,n} & h^R_{3m-2,n} & -h^I_{3m,n} & h^R_{3m,n} & 0 & 0 \\ h^R_{3m-1,n} & -h^I_{3m-1,n} & 0 & 0 & -h^R_{3m,n} & h^I_{3m,n} \\ h^I_{3m-1,n} & h^R_{3m-1,n} & 0 & 0 & -h^I_{3m,n} & -h^R_{3m,n} \\ h^R_{3m,n} & h^I_{3m,n} & h^R_{3m-2,n} & -h^I_{3m-2,n} & h^R_{3m-1,n} & h^I_{3m-1,n} \\ h^I_{3m,n} & -h^R_{3m,n} & h^I_{3m-2,n} & h^R_{3m-2,n} & h^I_{3m-1,n} & -h^R_{3m-1,n} \\ 0 & 0 & h^R_{3m-1,n} & h^I_{3m-1,n} & -h^R_{3m-2,n} & h^I_{3m-2,n} \\ 0 & 0 & h^I_{3m-1,n} & -h^R_{3m-1,n} & -h^I_{3m-2,n} & -h^R_{3m-2,n} \end{bmatrix}. \tag{6.22}$$

Lastly, the real-valued effective channel matrix for the rate-$\frac{3}{4}$ orthogonal code for four antennas is

$$\boldsymbol{\mathcal{H}}_{n,m} = \begin{bmatrix} h^R_{4m-3,n} & -h^I_{4m-3,n} & -h^R_{4m-1,n} & -h^I_{4m-1,n} & h^R_{4m,n} & h^I_{4m,n} \\ h^I_{4m-3,n} & h^R_{4m-3,n} & -h^I_{4m-1,n} & h^R_{4m-1,n} & h^I_{4m,n} & -h^R_{4m,n} \\ h^R_{4m-2,n} & -h^I_{4m-2,n} & -h^R_{4m,n} & h^I_{4m,n} & -h^R_{4m-1,n} & h^I_{4m-1,n} \\ h^I_{4m-2,n} & h^R_{4m-2,n} & -h^I_{4m,n} & -h^R_{4m,n} & -h^I_{4m-1,n} & -h^R_{4m-1,n} \\ h^R_{4m-1,n} & h^I_{4m-1,n} & h^R_{4m-3,n} & -h^I_{4m-3,n} & h^R_{4m-2,n} & h^I_{4m-2,n} \\ h^I_{4m-1,n} & -h^R_{4m-1,n} & h^I_{4m-3,n} & h^R_{4m-3,n} & h^I_{4m-2,n} & -h^R_{4m-2,n} \\ h^R_{4m,n} & h^I_{4m,n} & h^R_{4m-2,n} & h^I_{4m-2,n} & -h^R_{4m-3,n} & h^I_{4m-3,n} \\ h^I_{4m,n} & -h^R_{4m,n} & h^I_{4m-2,n} & -h^R_{4m-2,n} & -h^I_{4m-3,n} & -h^R_{4m-3,n} \end{bmatrix}. \tag{6.23}$$

As mentioned at the onset of the discussion, we will discuss the general form of the effective channel matrix of the embedded orthogonal space-time block code by drawing similarities to the effective channel matrix of the TAST code in (3.41). The real-valued effective channel matrix of the embedded orthogonal space-time block code of size $2T_1M_2N\times$

$2T_1M_2\mathcal{R}_1\mathcal{R}_2$ is

$$\check{\mathbf{H}} = \begin{bmatrix} \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0\bar{\mathbf{H}}_1\right) & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_1\right) & \cdots & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^{\mathcal{R}_2-1}\bar{\mathbf{H}}_1\right) \\ \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0\bar{\mathbf{H}}_2\right) & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_2\right) & \cdots & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^{\mathcal{R}_2-1}\bar{\mathbf{H}}_2\right) \\ \vdots & \vdots & \ddots & \vdots \\ \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0\bar{\mathbf{H}}_N\right) & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_N\right) & \cdots & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^{\mathcal{R}_2-1}\bar{\mathbf{H}}_N\right) \end{bmatrix} \left(\mathbf{I}_{\mathcal{R}_2} \otimes \bar{\mathbf{G}}\right)$$

$$= \begin{bmatrix} \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0\bar{\mathbf{H}}_1\right)\bar{\mathbf{G}} & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_1\right)\bar{\mathbf{G}} & \cdots & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^{\mathcal{R}_2-1}\bar{\mathbf{H}}_1\right)\bar{\mathbf{G}} \\ \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0\bar{\mathbf{H}}_2\right)\bar{\mathbf{G}} & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_2\right)\bar{\mathbf{G}} & \cdots & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^{\mathcal{R}_2-1}\bar{\mathbf{H}}_2\right)\bar{\mathbf{G}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0\bar{\mathbf{H}}_N\right)\bar{\mathbf{G}} & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_N\right)\bar{\mathbf{G}} & \cdots & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^{\mathcal{R}_2-1}\bar{\mathbf{H}}_N\right)\bar{\mathbf{G}} \end{bmatrix}, \quad (6.24)$$

where

- $\boldsymbol{\mathcal{J}}^\ell = \mathcal{F}_R\left(\mathbf{J}^\ell \otimes \mathbf{I}_{T_1}\right)$, $\ell \in \{0, 1, \cdots, \mathcal{R}_2 - 1\}$ and $\mathbf{J} = [\phi\mathbf{e}_{M_2}, \mathbf{e}_1, \cdots, \mathbf{e}_{M_2-1}]$
- $\bar{\mathbf{H}}_n = \begin{bmatrix} \boldsymbol{\mathcal{H}}_{n,1} \\ \boldsymbol{\mathcal{H}}_{n,2} \\ \vdots \\ \boldsymbol{\mathcal{H}}_{n,M_2-1} \\ \boldsymbol{\mathcal{H}}_{n,M_2} \end{bmatrix}$
- $\mathtt{blkdiag}(\bar{\mathbf{H}}_n)$ is the $2T_1M_2 \times 2T_1\mathcal{R}_1M_2$ *block* diagonal matrix with the $2T_1 \times 2T_1\mathcal{R}_1$ subblocks $\boldsymbol{\mathcal{H}}_{n,m}$ of $\bar{\mathbf{H}}_n$ on the diagonal
- $\bar{\mathbf{G}}$ is the $2M_2\mathcal{R}_1T_1 \times 2M_2\mathcal{R}_1T_1$ real-valued *effective* generator matrix that is related to the $M_2 \times M_2$ generator matrix $\mathbf{G}$ in (6.3) as follows

$$\bar{\mathbf{G}} = \mathcal{F}_R\left(\mathbf{G}\right) \otimes \mathbf{I}_{\mathcal{R}_1T_1}. \quad (6.25)$$

  Because $\mathbf{G}$ is real, $\mathcal{F}_R\left(\mathbf{G}\right) = \mathbf{G} \otimes \mathbf{I}_2$ and $\bar{\mathbf{G}}$ can be written as

$$\bar{\mathbf{G}} = (\mathbf{G} \otimes \mathbf{I}_2) \otimes \mathbf{I}_{\mathcal{R}_1T_1} = \mathbf{G} \otimes \mathbf{I}_{2\mathcal{R}_1T_1}. \quad (6.26)$$

The orthogonality embedding concept is not only seen in the construction of the embedded orthogonal space-time code, but it is also seen in the general form of the effective channel

matrix. The orthogonality embedding concept is evident by comparing (3.41) and (6.24). Specifically, we see that the fading coefficients $h_{m,n}$ in (3.41) have been replaced by the orthogonal effective channel matrices $\boldsymbol{\mathcal{H}}_{n,m}$ in (6.24).

In the construction of the embedded orthogonal space-time code, we remarked that if the ratio $\mathcal{R}/\mathcal{R}_1$ is not an integer, then the rate of the embedded orthogonal space-time code is higher than $\mathcal{R}$. Consequently, a rate-$\mathcal{R}$ embedded orthogonal code is obtained by puncturing the embedded codewords in the $\mathcal{R}_2$-th thread. When the ratio $\mathcal{R}/\mathcal{R}_1$ is not an integer, then the number of columns in the effective channel matrix is more than the number of transmitted symbols. To obtain the effective channel matrix with the appropriate number of columns, we simply delete the columns corresponding to the punctured symbols.

For clarity of exposition, we present the general form of the effective channel matrix of the rate-2 embedded orthogonal space-time code for four transmit and two receive antennas in the following example.

Example 6.4. Consider the transmission of the rate-2 embedded orthogonal space-time block code for four transmit antennas discussed in Example 6.1. The real-valued system model is given by (6.19) where the real-valued effective channel matrix in (6.24) is given by

$$\begin{aligned} \check{\mathbf{H}} &= \begin{bmatrix} \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0\bar{\mathbf{H}}_1\right) & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_1\right) \\ \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0\bar{\mathbf{H}}_2\right) & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_2\right) \end{bmatrix} \left(\mathbf{I}_2 \otimes \bar{\mathbf{G}}\right) \\ &= \begin{bmatrix} \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0\bar{\mathbf{H}}_1\right)\bar{\mathbf{G}} & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_1\right)\bar{\mathbf{G}} \\ \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0\bar{\mathbf{H}}_2\right)\bar{\mathbf{G}} & \mathtt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_2\right)\bar{\mathbf{G}} \end{bmatrix} \end{aligned} \tag{6.27}$$

where

- $\boldsymbol{\mathcal{J}}^0 = \mathbf{I}_8$ and $\boldsymbol{\mathcal{J}}^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \phi^R & -\phi^I & 0 & 0 & 0 & 0 & 0 & 0 \\ \phi^I & \phi^R & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \phi^R & -\phi^I & 0 & 0 & 0 & 0 \\ 0 & 0 & \phi^I & \phi^R & 0 & 0 & 0 & 0 \end{bmatrix}$
- $\bar{\mathbf{H}}_1 = \begin{bmatrix} \boldsymbol{\mathcal{H}}_{1,1} \\ \boldsymbol{\mathcal{H}}_{1,2} \end{bmatrix}$, $\bar{\mathbf{H}}_2 = \begin{bmatrix} \boldsymbol{\mathcal{H}}_{2,1} \\ \boldsymbol{\mathcal{H}}_{2,2} \end{bmatrix}$, and $\boldsymbol{\mathcal{H}}_{n,m}$ is given by (6.21)
- $\bar{\mathbf{G}} = (\mathbf{G} \otimes \mathbf{I}_4)$ and $\mathbf{G}$ is the 2×2 generator matrix given by (6.7)

Given the definitions of $\boldsymbol{\mathcal{J}}^\ell$ for $\ell \in \{1, 2\}$, $\bar{\mathbf{H}}_n$ for $n \in \{1, 2\}$ and $\mathbf{G}$, the real-valued effective channel matrix can be expressed in terms of the fading coefficients $h_{m,n}$, $m \in \{1, \cdots, 4\}$, $n \in \{1, 2\}$, as follows

$$\texttt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0 \bar{\mathbf{H}}_1\right) = \begin{bmatrix} h_{1,1}^R & -h_{1,1}^I & h_{2,1}^R & -h_{2,1}^I & 0 & 0 & 0 & 0 \\ h_{1,1}^I & h_{1,1}^R & h_{2,1}^I & h_{2,1}^R & 0 & 0 & 0 & 0 \\ h_{2,1}^R & h_{2,1}^I & -h_{1,1}^R & -h_{1,1}^I & 0 & 0 & 0 & 0 \\ h_{2,1}^I & -h_{2,1}^R & -h_{1,1}^I & h_{1,1}^R & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_{3,1}^R & -h_{3,1}^I & h_{4,1}^R & -h_{4,1}^I \\ 0 & 0 & 0 & 0 & h_{3,1}^I & h_{3,1}^R & h_{4,1}^I & h_{4,1}^R \\ 0 & 0 & 0 & 0 & h_{4,1}^R & h_{4,1}^I & -h_{3,1}^R & -h_{3,1}^I \\ 0 & 0 & 0 & 0 & h_{4,1}^I & -h_{4,1}^R & -h_{3,1}^I & h_{3,1}^R \end{bmatrix}, \quad (6.28)$$

$$\texttt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0\bar{\mathbf{H}}_2\right) = \begin{bmatrix} h_{1,2}^R & -h_{1,2}^I & h_{2,2}^R & -h_{2,2}^I & 0 & 0 & 0 & 0 \\ h_{1,2}^I & h_{1,2}^R & h_{2,2}^I & h_{2,2}^R & 0 & 0 & 0 & 0 \\ h_{2,2}^R & h_{2,2}^I & -h_{1,2}^R & -h_{1,2}^I & 0 & 0 & 0 & 0 \\ h_{2,2}^I & -h_{2,2}^R & -h_{1,2}^I & h_{1,2}^R & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_{3,2}^R & -h_{3,2}^I & h_{4,2}^R & -h_{4,2}^I \\ 0 & 0 & 0 & 0 & h_{3,2}^I & h_{3,2}^R & h_{4,2}^I & h_{4,2}^R \\ 0 & 0 & 0 & 0 & h_{4,2}^R & h_{4,2}^I & -h_{3,2}^R & -h_{3,2}^I \\ 0 & 0 & 0 & 0 & h_{4,2}^I & -h_{4,2}^R & -h_{3,2}^I & h_{3,2}^R \end{bmatrix}, \quad (6.29)$$

$\texttt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_1\right) =$

$$\begin{bmatrix} h_{3,1}^R & -h_{3,1}^I & h_{4,1}^R & -h_{4,1}^I & 0 & 0 & 0 & 0 \\ h_{3,1}^I & h_{3,1}^R & h_{4,1}^I & h_{4,1}^R & 0 & 0 & 0 & 0 \\ h_{4,1}^R & h_{4,1}^I & -h_{3,1}^R & -h_{3,1}^I & 0 & 0 & 0 & 0 \\ h_{4,1}^I & -h_{4,1}^R & -h_{3,1}^I & h_{3,1}^R & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \phi^R h_{1,1}^R - \phi^I h_{1,1}^I & -\phi^R h_{1,1}^I - \phi^I h_{1,1}^R & \phi^R h_{2,1}^R - \phi^I h_{2,1}^I & -\phi^R h_{2,1}^I - \phi^I h_{2,1}^R \\ 0 & 0 & 0 & 0 & \phi^I h_{1,1}^R + \phi^R h_{1,1}^I & \phi^R h_{1,1}^R - \phi^I h_{1,1}^I & \phi^R h_{2,1}^I + \phi^I h_{2,1}^R & \phi^R h_{2,1}^R - \phi^I h_{2,1}^I \\ 0 & 0 & 0 & 0 & \phi^R h_{2,1}^R - \phi^I h_{2,1}^I & \phi^R h_{2,1}^I + \phi^I h_{2,1}^R & -\phi^R h_{1,1}^R + \phi^I h_{1,1}^I & -\phi^R h_{1,1}^I - \phi^I h_{1,1}^R \\ 0 & 0 & 0 & 0 & \phi^R h_{2,1}^I + \phi^I h_{2,1}^R & -\phi^R h_{2,1}^R + \phi^I h_{2,1}^I & -\phi^R h_{1,1}^I - \phi^I h_{1,1}^R & \phi^R h_{1,1}^R - \phi^I h_{1,1}^I \end{bmatrix}, \quad (6.30)$$

$\texttt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1\bar{\mathbf{H}}_2\right) =$

$$\begin{bmatrix} h_{3,2}^R & -h_{3,2}^I & h_{4,2}^R & -h_{4,2}^I & 0 & 0 & 0 & 0 \\ h_{3,2}^I & h_{3,2}^R & h_{4,2}^I & h_{4,2}^R & 0 & 0 & 0 & 0 \\ h_{4,2}^R & h_{4,2}^I & -h_{3,2}^R & -h_{3,2}^I & 0 & 0 & 0 & 0 \\ h_{4,2}^I & -h_{4,2}^R & -h_{3,2}^I & h_{3,2}^R & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \phi^R h_{1,2}^R - \phi^I h_{1,2}^I & -\phi^R h_{1,2}^I - \phi^I h_{1,2}^R & \phi^R h_{2,2}^R - \phi^I h_{2,2}^I & -\phi^R h_{2,2}^I - \phi^I h_{2,2}^R \\ 0 & 0 & 0 & 0 & \phi^I h_{1,2}^R + \phi^R h_{1,2}^I & \phi^R h_{1,2}^R - \phi^I h_{1,2}^I & \phi^R h_{2,2}^I + \phi^I h_{2,2}^R & \phi^R h_{2,2}^R - \phi^I h_{2,2}^I \\ 0 & 0 & 0 & 0 & \phi^R h_{2,2}^R - \phi^I h_{2,2}^I & \phi^R h_{2,2}^I + \phi^I h_{2,2}^R & -\phi^R h_{1,2}^R + \phi^I h_{1,2}^I & -\phi^R h_{1,2}^I - \phi^I h_{1,2}^R \\ 0 & 0 & 0 & 0 & \phi^R h_{2,2}^I + \phi^I h_{2,2}^R & -\phi^R h_{2,2}^R + \phi^I h_{2,2}^I & -\phi^R h_{1,2}^I - \phi^I h_{1,2}^R & \phi^R h_{1,2}^R - \phi^I h_{1,2}^I \end{bmatrix}, \quad (6.31)$$

and

$$\bar{\mathbf{G}} = \begin{bmatrix} \alpha_1 & 0 & 0 & 0 & \alpha_2 & 0 & 0 & 0 \\ 0 & \alpha_1 & 0 & 0 & 0 & \alpha_2 & 0 & 0 \\ 0 & 0 & \alpha_1 & 0 & 0 & 0 & \alpha_2 & 0 \\ 0 & 0 & 0 & \alpha_1 & 0 & 0 & 0 & \alpha_2 \\ -\alpha_2 & 0 & 0 & 0 & \alpha_1 & 0 & 0 & 0 \\ 0 & -\alpha_2 & 0 & 0 & 0 & \alpha_1 & 0 & 0 \\ 0 & 0 & -\alpha_2 & 0 & 0 & 0 & \alpha_1 & 0 \\ 0 & 0 & 0 & -\alpha_2 & 0 & 0 & 0 & \alpha_1 \end{bmatrix}. \tag{6.32}$$

□

We next discuss the properties of the effective channel matrix of the embedded orthogonal space-time block codes that lead to a reduction in the worst-case ML decoding complexity.

6.2.2 Properties of The Effective Channel Matrix of the Embedded Orthogonal Space-Time Block Code

The decoding complexity of the embedded orthogonal space-time block code depends on the number of transmit antennas, rate $\mathcal{R}$, and the choice of the embedded orthogonal design. We will first discuss the case when $\mathcal{R} = \mathcal{R}_1$ and show that the embedded orthogonal space-time block code is separable. We discuss the separability of the embedded orthogonal design in terms of the properties of the $\mathbf{R}$ matrix in the QR decomposition of the effective channel matrix in the following lemma.

Lemma 6.2. *The embedded orthogonal space-time block code for $M = M_1 M_2$ transmit antennas embedding $\boldsymbol{\mathcal{G}}(M_1, \mathcal{R}_1)$ is η-group decodable when $\mathcal{R} = \mathcal{R}_1$, where $\eta = 2\mathcal{R}_1 T_1$. Furthermore, the worst-case ML decoding complexity is $\mathcal{O}\left(q^{\frac{M/M_1-1}{2}}\right) = \mathcal{O}\left(q^{\frac{M_2-1}{2}}\right)$.*

Proof. See Appendix E. □

We summarize the separability results and worst-case ML decoding complexity of the embedded orthogonal space-time block code when we embed the Alamouti code and the rate-$\frac{3}{4}$ orthogonal designs for three and four transmit antennas in the following three corollaries.

Corollary 6.2. *When the embedded code is the Alamouti space-time block code $\boldsymbol{\mathcal{G}}(2,1)$, then the rate $\mathcal{R}=1$ embedded orthogonal space-time block code for $M=M_1M_2=2M_2$ transmit antennas is 4-group decodable, with a worst-case ML decoding complexity of $\mathcal{O}\left(q^{\frac{M/2-1}{2}}\right)$.*

Corollary 6.3. *When the embedded code is the rate-$\frac{3}{4}$ orthogonal space-time code $\boldsymbol{\mathcal{G}}(3,\frac{3}{4})$ in (2.20) for three transmit antennas, then the rate $\mathcal{R}=\frac{3}{4}$ embedded orthogonal space-time block code for $M=M_1M_2=3M_2$ transmit antennas is 6-group decodable, with a worst-case ML decoding complexity of $\mathcal{O}\left(q^{\frac{M/3-1}{2}}\right)$.*

Corollary 6.4. *When the embedded code is the rate-$\frac{3}{4}$ orthogonal space-time code $\boldsymbol{\mathcal{G}}(4,\frac{3}{4})$ in (2.20) for four transmit antennas, then the rate $\mathcal{R}=\frac{3}{4}$ embedded orthogonal space-time block code for $M=M_1M_2=4M_2$ transmit antennas is 6-group decodable, with a worst-case ML decoding complexity of $\mathcal{O}\left(q^{\frac{M/4-1}{2}}\right)$.*

Having discussed the separability of the embedded orthogonal code for $\mathcal{R}=\mathcal{R}_1$, we next discuss the case of $\mathcal{R}>\mathcal{R}_1$. Specifically, we prove that the embedded orthogonal space-time block code is not separable for $\mathcal{R}>\mathcal{R}_1$ in the following lemma.

Lemma 6.3. *The embedded orthogonal space-time block code is not separable for $\mathcal{R}>\mathcal{R}_1$.*

Proof. See Appendix F. □

Corollary 6.5. *When the embedded code is the Alamouti space-time block code, then the rate $\mathcal{R}>1$ embedded orthogonal space-time block code is not separable.*

Corollary 6.6. *When the embedded code is the rate-$\frac{3}{4}$ orthogonal design for three or four transmit antennas, then the rate $\mathcal{R}>\frac{3}{4}$ embedded orthogonal space-time block code is not separable.*

Using lemma 6.2 and lemma 6.3, we arrive at the main theorem regarding the worst-case ML decoding complexity of the embedded orthogonal space-time block codes.

Theorem 6.1. *The worst-case maximum-likelihood decoding complexity of a rate-$\mathcal{R}$ embedded orthogonal space-time block code with $\boldsymbol{\mathcal{G}}(M_1,\mathcal{R}_1)$ for $M=M_1M_2$ transmit antennas is $\mathcal{O}\left(q^{(\mathcal{R}-\mathcal{R}_1)T_1M_2+\frac{M_2-1}{2}}\right)$.*

Proof. We will use a similar technique to theorem 3.1. The worst-case ML decoding complexity of the rate-$\mathcal{R}$ embedded orthogonal space-time block code is the worst-case decoding complexity of the first $\mathcal{R}-\mathcal{R}_1$ layers, multiplied by the worst-case ML decoding complexity of the last $\mathcal{R}=\mathcal{R}_1$ layer. Therefore, the worst-case decoding of the rate-$\mathcal{R}$ embedded orthogonal code is $\mathcal{O}\left(q^{(\mathcal{R}-\mathcal{R}_1)T}\right)\times\mathcal{O}\left(q^{\frac{M_2-1}{2}}\right)$. The first term is the worst-case ML decoding complexity of the first $\mathcal{R}-\mathcal{R}_1$ layers, and the last term is the decoding complexity of the last layer when $\mathcal{R}=\mathcal{R}_1$, as proven in lemma 6.2. Consequently, the worst-case ML decoding complexity is $\mathcal{O}\left(q^{(\mathcal{R}-\mathcal{R}_1)T_1M_2+\frac{M_2-1}{2}}\right)$, where we simply substituted for the definition of $T=T_1M_2$. □

The worst-case ML decoding complexity for the perfect, threaded algebraic, quasiorthogonal, semi-orthogonal algebraic and proposed embedded orthogonal space-time block codes is summarized in Table 6.2 for up to eight transmit antennas. As can be seen from the Table 6.2, the embedded orthogonal space-time block code has the lowest decoding complexity compared to these families of space-time block codes.

6.2.3 Efficient ML Decoding of the Embedded Orthogonal Space-Time Codes

In Theorem 6.1, we proved that the embedded orthogonal space-time block code allows for a reduction in decoding complexity compared to an exhaustive-search decoder. In this section, we discuss an efficient ML decoder with low average complexity and a worst-case complexity of $\mathcal{O}\left(q^{(\mathcal{R}-\mathcal{R}_1)T_1M_2+\frac{M_2-1}{2}}\right)$. We begin our discussion with the case of $\mathcal{R}=\mathcal{R}_1$. The efficient ML decoder for $\mathcal{R}>\mathcal{R}_1$ utilizes the decoder for $\mathcal{R}=\mathcal{R}_1$.

In Lemma 6.2, we proved that the embedded orthogonal code for $\mathcal{R}=\mathcal{R}_1$ is η-group decodable, with a worst-case decoding complexity of $\mathcal{O}\left(q^{\frac{M_2-1}{2}}\right)=\mathcal{O}\left(\sqrt{q}^{M_2-1}\right)$, where $\eta=2\mathcal{R}_1T_1$. Therefore, the embedded orthogonal code can be decoded with η real-valued sphere decoders, each with M_2 levels. The pseudocode of a K-level real-valued sphere decoder is shown in Figure 6.1. The sphere decoder in Figure 6.1 is a generalization of the sphere decoder in Figure 3.4. Apart from the fact that the sphere decoder in Figure 6.1 has K levels compared to only three levels in Figure 3.4, the only other difference between the two decoders is the use of a real PAM alphabet $\mathcal{A}^R$ with cardinality $\sqrt{q}$ in Figure 6.1. In

Table 6.2: Worst-Case ML Decoding Complexity for Several Space-Time Block Codes.

M	$\mathcal{R}$	Perfect	TAST	Quasiorthogonal	SAST	Proposed EOS
2	1	$\mathcal{O}\left(q^{0.5}\right)$	$\mathcal{O}\left(q^{0.5}\right)$	-	$\mathcal{O}\left(1\right)$	$\mathcal{O}\left(1\right)$
4	0.75	-	-	-	-	$\mathcal{O}\left(1\right)^{a}$
	1	$\mathcal{O}\left(q^{1.5}\right)$	$\mathcal{O}\left(q^{1.5}\right)$	$\mathcal{O}\left(q\right)$	$\mathcal{O}\left(q^{0.5}\right)$	$\mathcal{O}\left(q^{0.5}\right)$
	1.5	$\mathcal{O}\left(q^{3.5}\right)$	$\mathcal{O}\left(q^{3.5}\right)$	-	-	$\mathcal{O}\left(q^{2.5}\right)$
	2	$\mathcal{O}\left(q^{5.5}\right)$	$\mathcal{O}\left(q^{5.5}\right)$	-	-	$\mathcal{O}\left(q^{4.5}\right)$
6	0.75	-	-	-	-	$\mathcal{O}\left(q^{0.5}\right)^{b}$
	1	$\mathcal{O}\left(q^{5}\right)$	$\mathcal{O}\left(q^{2.5}\right)$	$\mathcal{O}\left(q^{3}\right)$	$\mathcal{O}\left(q\right)$	$\mathcal{O}\left(q\right)$
	1.5	$\mathcal{O}\left(q^{8}\right)$	$\mathcal{O}\left(q^{5.5}\right)$	-	-	$\mathcal{O}\left(q^{4}\right)$
						$\mathcal{O}\left(q^{6.5}\right)^{c}$
	2	$\mathcal{O}\left(q^{11}\right)$	$\mathcal{O}\left(q^{8.5}\right)$	-	-	$\mathcal{O}\left(q^{7}\right)$
	3	$\mathcal{O}\left(q^{17}\right)$	$\mathcal{O}\left(q^{14.5}\right)$	-	-	$\mathcal{O}\left(q^{13}\right)$
8	0.75	-	-	-	-	$\mathcal{O}\left(q^{0.5}\right)^{d}$
	1	$\mathcal{O}\left(q^{3.5}\right)$	$\mathcal{O}\left(q^{3.5}\right)$	$\mathcal{O}\left(q^{3}\right)$	$\mathcal{O}\left(q^{1.5}\right)$	$\mathcal{O}\left(q^{1.5}\right)$
	1.5	$\mathcal{O}\left(q^{7.5}\right)$	$\mathcal{O}\left(q^{7.5}\right)$	-	-	$\mathcal{O}\left(q^{5.5}\right)$
						$\mathcal{O}\left(q^{6.5}\right)^{e}$
	2	$\mathcal{O}\left(q^{11.5}\right)$	$\mathcal{O}\left(q^{11.5}\right)$	-	-	$\mathcal{O}\left(q^{9.5}\right)$
	3	$\mathcal{O}\left(q^{19.5}\right)$	$\mathcal{O}\left(q^{19.5}\right)$	-	-	$\mathcal{O}\left(q^{17.5}\right)$
	4	$\mathcal{O}\left(q^{27.5}\right)$	$\mathcal{O}\left(q^{27.5}\right)$	-	-	$\mathcal{O}\left(q^{25.5}\right)$

[a,d,e] embedded $\boldsymbol{\mathcal{G}}\left(4, \frac{3}{4}\right)$
[b,c] embedded $\boldsymbol{\mathcal{G}}\left(3, \frac{3}{4}\right)$

Figure 6.1, we introduced the notation $\mathbf{R}_{a:b,c:d}$ to denote the sub-matrix consisting of rows a through b and columns c through d of the matrix $\mathbf{R}$. We next discuss the efficient ML decoder of the embedded orthogonal space-time code for $\mathcal{R} = \mathcal{R}_1$ shown in Figure 6.2.

The sphere decoder in Figure 6.2, termed `EOS Multigroup Decoder` is based on the fact that the embedded orthogonal space-time block code is η-group decodable, with each group containing M_2 real symbols. The first two lines are initializations, where we perform a QR decomposition of the effective channel matrix in line 1 and form the intermediate variable $\mathbf{z}$ in line 2. The next five lines (line 3 through line 7) represent a loop that calls a real-valued

Algorithm: $[\hat{P}, \hat{\mathbf{x}}] = \texttt{Sphere Decoder}(\check{\mathbf{y}}, \check{\mathbf{H}}, K)$

$[\mathbf{Q}, \mathbf{R}] = \texttt{QR decomposition}(\check{\mathbf{H}})$
$\mathbf{z} = \mathbf{Q}^\top \check{\mathbf{y}}$
$\hat{P} = \infty$
$[P_K, \Pi_K] = \texttt{sort}_{a \in \mathcal{A}^R}(|z_K - r_{K,K} a|^2)$
for n_K from 1 to $\sqrt{q}$ **do**
  if $P_K(n_K) > \hat{P}$, **break**, **end**
  $x_K = \mathcal{A}^R(\Pi_K(n_K))$
  $v = z_{K-1} - \mathbf{R}_{K-1,K} x_K$
  $[P_{K-1}, \Pi_{K-1}] = \texttt{sort}_{a \in \mathcal{A}^R}(|v - r_{K-1,K-1} a|^2)$
  for n_{K-1} from 1 to $\sqrt{q}$ **do**
    if $(P_{K-1}(n_{K-1}) + P_K(n_K)) > \hat{P}$, **break**, **end**
    $x_{K-1} = \mathcal{A}^R(\Pi_{K-1}(n_{K-1}))$
    $v = z_{K-2} - \mathbf{R}_{K-2,K-1:K} [x_{K-1}, x_K]^\top$
    $[P_{K-2}, \Pi_{K-2}] = \texttt{sort}_{a \in \mathcal{A}^R}(|v - r_{K-2,K-2} a|^2)$
    $\vdots$
    for n_2 from 1 to $\sqrt{q}$ **do**
      if $(P_2(n_2) + \cdots + P_K(n_K)) > \hat{P}$, **break**, **end**
      $x_2 = \mathcal{A}^R(\Pi_2(n_2))$
      $v = z_1 - \mathbf{R}_{1,2:K} [x_2, \cdots, x_K]^\top$
      $x_1 = Q(v / r_{1,1})$
      $P_1 = |v - r_{1,1} x_1|^2$
      $P = P_1 + \sum_{k=2}^{K} P_k(n_k)$
      if $P < \hat{P}$ **then**
        $\hat{P} = P$
        $\hat{\mathbf{x}} = [x_1, x_2, \cdots, x_K]$
      end
    end
    $\vdots$
  end
end

Figure 6.1: Real-valued sphere decoder with K tree levels.

Algorithm: $[\bar{P}, \mathbf{x}] =$ `EOS Multigroup Decoder`$(\check{\mathbf{y}}, \check{\mathbf{H}}, \eta, M_2)$

1 $[\mathbf{Q}, \mathbf{R}] =$ `QR decomposition`$(\check{\mathbf{H}})$
2 $\mathbf{z} = \mathbf{Q}^\top \check{\mathbf{y}}$
3 **for** n from 1 to η **do**
4 $\quad \bar{\mathbf{z}} = [z_n, z_{n+\eta}, \cdots, z_{n+(M_2-1)\eta}]$
5 $\quad$ Form $\bar{\mathbf{R}}$ from $\mathbf{R}$ according to (E.5)
6 $\quad [P_n, [x_n, x_{n+\eta}, \cdots, x_{n+(M_2-1)\eta}]] =$ `Sphere Decoder`$(\bar{\mathbf{z}}, \bar{\mathbf{R}}, M_2)$
7 **end**
8 $\bar{P} = \sum_{n=1}^{\eta} P_n$
9 $\mathbf{x} = [x_1, \cdots, x_{\eta M_2}]$

Figure 6.2: Pseudocode of a fast ML decoder for the embedded orthogonal space-time block codes for $\mathcal{R} = \mathcal{R}_1$.

sphere decoder with M_2 levels. Specifically, the real-valued sphere decoder in Figure 6.1 is called in line 6 with an appropriate received vector $\bar{\mathbf{z}}$ (line 4), an appropriate effective channel matrix $\bar{\mathbf{R}}$ (line 5), and the parameter M_2 for the number of levels in the decoder. We will discuss some important implications of the loop as written in the pseudocode shortly. First, however, we proceed with the rest of the decoding algorithm. The path metric in line 8 is the sum of path metrics from the η decoders. Finally, the ML decision is formed in line 9 from the ML decisions of the η decoders.

We now discuss two implications regarding the loop in line 3 through line 7. First, we have chosen to write a loop which executes the algorithm sequentially. We note, however, that the loop can be *unrolled*, such that the `Sphere Decoder` algorithm executes in parallel in a hardware implementation. On one extreme, the hardware can have as many as η instances of the `Sphere Decoder` block running in parallel. This hardware implementation has the largest area, but it also has the lowest decoding latency and highest throughput. On the other extreme, the hardware can have one instance of the `Sphere Decoder` block, and it executes η times, similar to what is shown in the pseudocode. This hardware implementation has the smallest area, but it also has the highest decoding latency and lowest throughput. In a practical implementation, hardware designers can choose between the two extremes depending on the area, power dissipation, latency and throughput constraints of the system.

Second, it is not necessary to perform a QR decomposition inside the `Sphere Decoder`

function when it is being called inside the loop. This is because the matrix $\bar{\mathbf{R}}$ is already an $M_2 \times M_2$ upper triangular matrix. In a practical implementation, the QR decomposition might be performed outside of the decoding function. We perform the QR decomposition inside the function for the sake of self completeness.

Having discussed an efficient ML decoder for $\mathcal{R} = \mathcal{R}_1$ for the embedded orthogonal space-time code, we now present an efficient ML decoder for $\mathcal{R} > \mathcal{R}_1$. The basic idea of the decoding algorithm follows from the proof of Theorem 6.1. Specifically, the algorithm consists of two parts. The first part decides on the best candidate symbol vector for the first $\mathcal{R}-\mathcal{R}_1$ layers, while the second part finds the best candidate symbol vector for the last $\mathcal{R} = \mathcal{R}_1$ layer, after canceling the interference from the first $\mathcal{R} - \mathcal{R}_1$ layers. As might be expected, the second part of the decoding algorithm that finds the best candidate symbol vector for the last $\mathcal{R} = \mathcal{R}_1$ layer is simply the `EOS Multigroup Decoder` function in Figure 6.2. The pseudocode of an efficient ML decoder for $\mathcal{R} > \mathcal{R}_1$, termed `EOS Decoder`, is shown in Figure 6.3. We next discuss the pseudocode in detail.

The first four lines are initializations that include a QR decomposition of the effective channel matrix (line 1), forming the variable $\mathbf{z}$ (line 2), determining the number of symbols in the last $\mathcal{R} = \mathcal{R}_1$ layer (line 3) and initializing the sphere radius (line 4). The remaining part of the algorithm can be interpreted as finding the candidate vector symbol for the first $2K - L$ real symbols, where $L = \eta M_2$, and for each candidate vector, we find the last L real symbols using the `Sphere Multigroup Decoder`. Specifically, line 5 through line 24 find the candidate symbol vector $[x_{L+1}, x_{L+2}, \cdots, x_{2K}]$. In fact, lines 5 through 24 represent a real-valued sphere decoder with $2K - L$ levels. We cancel the interference from the symbols $[x_{L+1}, x_{L+2}, \cdots, x_{2K}]$ in line 25, and then we find the candidate symbol vector $[x_1, x_2, \cdots, x_L]$ in line 26. The total path metric is computed in line 27. Sphere radius update (line 29) and best candidate vector update (line 30) occur if the current path metric is smaller than the sphere radius in line 28. We end the discussion regarding the pseudocode by noting that the QR decomposition performed inside the `EOS Multigroup Decoder` is unnecessary since the effective channel matrix being passed as an input argument is already an $L \times L$ upper triangular matrix. Again, we have chosen to include the QR decomposition

Algorithm: $[\hat{P}, \hat{\mathbf{x}}] = \texttt{EOS Decoder}(\check{\mathbf{y}}, \check{\mathbf{H}}, K, \eta, M_2)$

$[\mathbf{Q}, \mathbf{R}] = \texttt{QR decomposition}(\check{\mathbf{H}})$
$\mathbf{z} = \mathbf{Q}^{\top}\check{\mathbf{y}}$
$L = \eta M_2$
$\hat{P} = \infty$
$[P_{2K}, \Pi_{2K}] = \texttt{sort}_{a \in \mathcal{A}^R}(|z_{2K} - r_{2K,2K}a|^2)$
for n_{2K} from 1 to $\sqrt{q}$ **do**
  if $P_{2K}(n_{2K}) > \hat{P}$, **break**, **end**
  $x_{2K} = \mathcal{A}^R(\Pi_{2K}(n_{2K}))$
  $v = z_{2K-1} - \mathbf{R}_{2K-1,2K}x_{2K}$
  $[P_{2K-1}, \Pi_{2K-1}] = \texttt{sort}_{a \in \mathcal{A}^R}(|v - r_{2K-1,2K-1}a|^2)$
  for n_{2K-1} from 1 to $\sqrt{q}$ **do**
    if $(P_{2K-1}(n_{2K-1}) + P_{2K}(n_{2K})) > \hat{P}$, **break**, **end**
    $x_{2K-1} = \mathcal{A}^R(\Pi_{2K-1}(n_{2K-1}))$
    $v = z_{2K-2} - \mathbf{R}_{2K-2,2K-1:2K}[x_{2K-1}, x_{2K}]^{\top}$
    $[P_{2K-2}, \Pi_{2K-2}] = \texttt{sort}_{a \in \mathcal{A}^R}(|v - r_{2K-2,2K-2}a|^2)$
    $\vdots$
    for n_{L+2} from 1 to $\sqrt{q}$ **do**
      if $(P_{L+2}(n_{L+2}) + \cdots + P_{2K}(n_{2K})) > \hat{P}$, **break**, **end**
      $x_{L+2} = \mathcal{A}^R(\Pi_{L+2}(n_{L+2}))$
      $v = z_{L+1} - \mathbf{R}_{L+1,L+2:2K}[x_{L+2}, \cdots, x_{2K}]^{\top}$
      $[P_{L+1}, \Pi_{L+1}] = \texttt{sort}_{a \in \mathcal{A}^R}(|v - r_{L+1,L+1}a|^2)$
      for n_{L+1} from 1 to $\sqrt{q}$ **do**
        if $(P_{L+1}(n_{L+1}) + \cdots + P_{2K}(n_{2K})) > \hat{P}$, **break**, **end**
        $x_{L+1} = \mathcal{A}^R(\Pi_{L+1}(n_{L+1}))$
        $\mathbf{v} = [z_1, \cdots, z_L]^{\top} - \mathbf{R}_{1:L,L+1:2K}[x_{L+1}, \cdots, x_{2K}]^{\top}$
        $[\bar{P}, [x_1, x_2, \cdots, x_L]] = \texttt{EOS Multigroup Decoder}(\mathbf{v}, \mathbf{R}_{1:L,1:L}, \eta, M_2)$
        $\hat{P} = \sum_{k=L+1}^{2K} P_k(n_k) + \bar{P}$
        if $P < \hat{P}$ **then**
          $\hat{P} = P$
          $\hat{\mathbf{x}} = [x_1, x_2, \cdots, x_{2K}]$
        end
      end
    end
    $\vdots$
  end
end

Figure 6.3: Pseudocode of a fast ML decoder for the embedded orthogonal space-time block codes for $\mathcal{R} > \mathcal{R}_1$.

inside the function for the sake of self completeness.

Having discussed an efficient decoder for the embedded orthogonal space-time codes, we note that the basic ideas of the decoding algorithm can be used to decode the TAST and perfect codes with real generator matrices and QAM alphabet. This is because the TAST and perfect codes are 2-group decodable, with each group containing M real symbols for $\mathcal{R} = 1$. Therefore, for $\mathcal{R} = 1$, we can use two real-valued sphere decoders, each having M levels. By comparison, the embedded orthogonal code uses η real-valued sphere decoders, $\eta \in \{4, 6\}$, each having $M_2 < M$ real symbols. The decoding for $\mathcal{R} > 1$ for the perfect and TAST codes follows similarly to the decoding of the embedded orthogonal codes for $\mathcal{R} > \mathcal{R}_1$. Specifically, the decoding algorithm also consists of two parts. The first part decides on the best candidate symbol vector for the first $\mathcal{R} - 1$ layers, while the second part finds the best candidate symbol vector for the last $\mathcal{R} = 1$ layer, after canceling the interference from the first $\mathcal{R} - 1$ layers.

6.3 Numerical Results

In this section, we present simulation results for up to six transmit antennas. The channel was modeled using (2.2) with quasistatic i.i.d. Rayleigh fading, with constant coefficients within each codeword block, but independent fading from block to block. The modulation alphabet was QAM for all the simulated space-time codes except for perfect codes for six transmit antennas, where the modulation alphabet is HEX. We show simulation results for $q = 4$, $q = 16$ and $q = 64$, respectively. We next comment on the different efficient ML decoders used for the different families of space-time block codes.

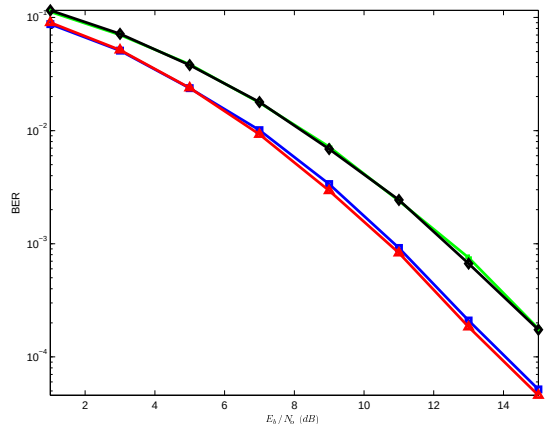
The embedded orthogonal code was decoded following the pseudocode in Figure 6.2 and Figure 6.3. Similarly, the perfect and TAST codes with QAM alphabet were decoded using similar pseudocode to Figures 6.2 and Figure 6.3 as discussed at the end of Section 6.2.3. The perfect code with HEX alphabet was decoded using a complex-valued sphere decoder, similar to the real-valued sphere decoder in Figure 6.1. The quasiorthogonal space-time block code, which is 2-group decodable was decoded using a similar pseudocode to Figure 6.2. We omit any mention or comparison with DAST code since the $\mathcal{R} = 1$ TAST code

reduces to the DAST code. Similarly, we omit simulation results for SAST codes since the embedded orthogonal space-time block code subsumes the SAST code as a special case for $\mathcal{R} = 1$. Finally, we omit simulation results for the single-symbol decodable space-time code since it is only defined for $M = 4$, in which case, it is also equivalent to the embedded orthogonal code for $M = 4$ and $\mathcal{R} = 1$.

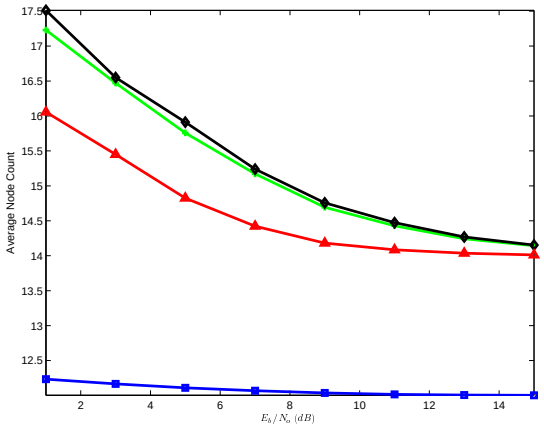
We start with $\mathcal{R} = 1$ for $M = 4$ and $M = 6$ transmit antennas shown in Figure 6.4 and Figure 6.5, respectively. In both figures, we compare complexity and performance for quasiorthogonal, TAST, perfect and embedded orthogonal space-time block codes.

In Figure 6.4, we see that quasiorthogonal codes and embedded orthogonal codes have the lowest error probability and lowest decoding complexity, respectively, for all spectral efficiencies. Therefore, there is not a single space-time code that is simultaneously better performing and lower in decoding complexity. This leads us to a complexity-performance tradeoff. Specifically, for a given SNR value at the receiver for which the best performing code achieves a target error probability of 10^{-3}, we determine the decoding complexity and additional SNR required by the other codes to achieve the same error probability. We summarize the complexity-performance tradeoff in Table 6.3. For example, for SNR = 10.7 dB at which the quasiorthogonal space-time code achieves an error probability of 10^{-3} for $q = 4$, we tabulate the additional SNR (i.e., SNR penalty) required by the embedded orthogonal, TAST and perfect space-time codes to achieve the same error probability. Furthermore, we tabulate the average complexity, measured by the average number of nodes visited in the tree search to reach the ML decision.

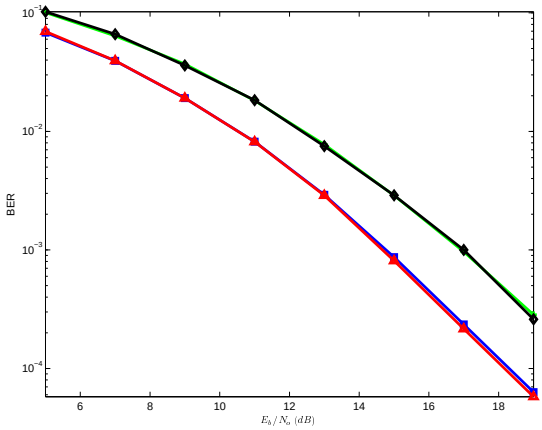
From Table 6.3, we see that quasiorthogonal and embedded orthogonal designs are favorable in terms of error rate performance and average complexity compared to TAST and perfect codes. For example, for $q = 64$, the perfect code requires an additional 2.55 and 2.50 dB to achieve the same error rate as quasiorthogonal and embedded orthogonal codes, respectively. Furthermore, the perfect code is 3% and 18% more complex than quasiorthogonal and embedded orthogonal codes, respectively. Embedded orthogonal codes are within 0.15 dB of quasiorthogonal codes in terms of error rate performance, and they are 15% less complex in terms of the average number of nodes visited. Furthermore, the



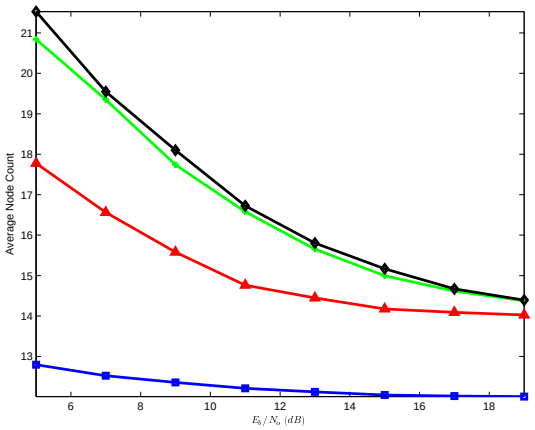
(a) Performance for $q = 4$



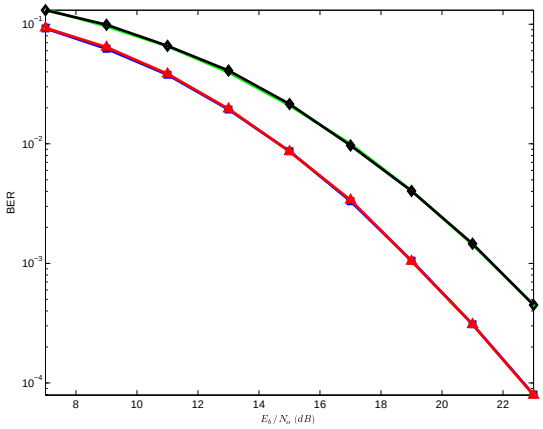
(b) Complexity for $q = 4$



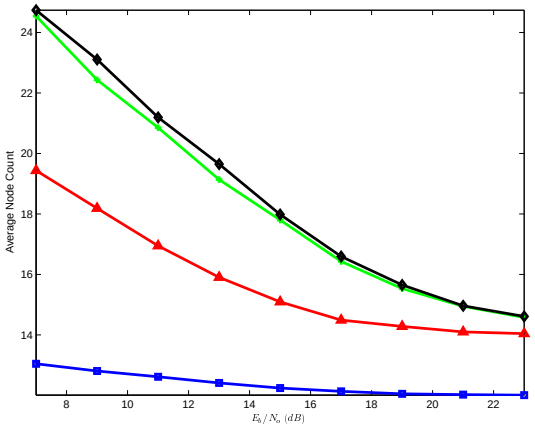
(c) Performance for $q = 16$



(d) Complexity for $q = 16$



(e) Performance for $q = 64$



(f) Complexity for $q = 64$

Figure 6.4: Performance and complexity comparison for $M = 4$, $\mathcal{R} = 1$ and $N = 1$ for quasiorthogonal ($\triangle$), TAST ($+$), perfect ($\diamond$) and embedded orthogonal space-time ($\square$) block codes.

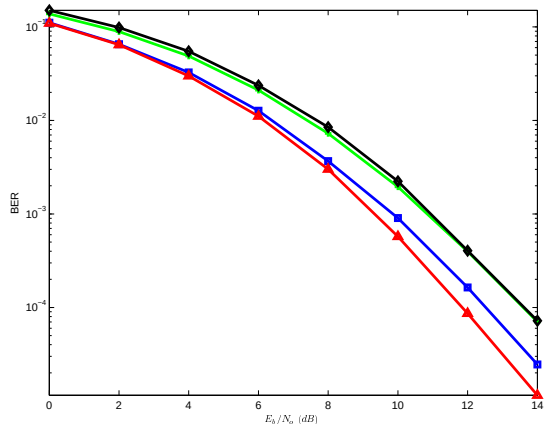
Table 6.3: Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 4$, $N = 1$, and $\mathcal{R} = 1$. Average Complexity and SNR Penalty are Relative to 10.7, 14.7 and 19.05 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively.

Code	Average Complexity			SNR Penalty (dB)		
	$q = 4$	$q = 16$	$q = 64$	$q = 4$	$q = 16$	$q = 64$
Quasiorthogonal	14.12	14.22	14.28	0	0	0
Embedded Orthogonal	12.02	12.05	12.06	0.15	0.08	0.02
Perfect	14.52	15.25	15.65	1.66	2.27	2.55
TAST	14.47	15.1	15.53	1.70	2.33	2.57

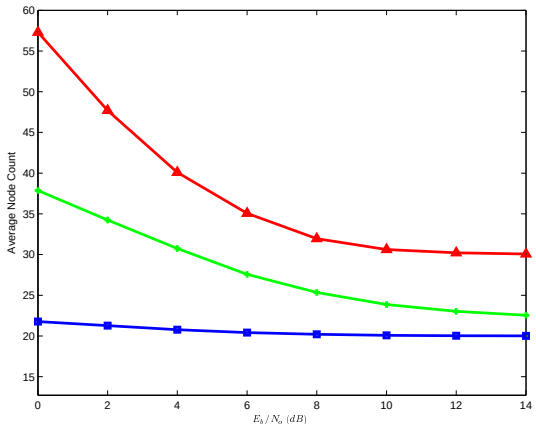
worst-case ML complexity of the embedded orthogonal code is 50%, 75% and 87% lower than that of quasiorthogonal codes for $q = 4$, $q = 16$ and $q = 64$, respectively.

The error rate performance and complexity for $\mathcal{R} = 1$ and $M = 6$ are shown in Figure 6.5. Because the real and imaginary parts in the HEX alphabet are not separable, we used the complex-valued sphere decoder for the perfect codes. For the remaining codes, however, we used the real-valued sphere decoder. In comparing the complex-valued and real-valued sphere decoders, one might be tempted to simply count each complex node as two real nodes. This comparison, however, is inaccurate. To illustrate that, consider a two-level complex sphere decoder and the corresponding four-level real-valued decoder. The minimum number of nodes visited by the complex-valued and real-valued decoders is 3 and 7, respectively. Hence, simply counting each complex node as two real nodes is unfair. Furthermore, the two-level complex-valued sphere decoder has 16+256=272 nodes, while the real-valued decoder has 4+16+64+256=340 nodes. Therefore, we will omit the complexity results for the perfect codes except for $q = 64$ to convince the reader that perfect codes have the highest decoding complexity. As can be seen from Figure 5(b), the perfect codes have the highest average node count even we are counting a complex node as one real node. We summarize the complexity-performance tradeoff in Table 6.4.

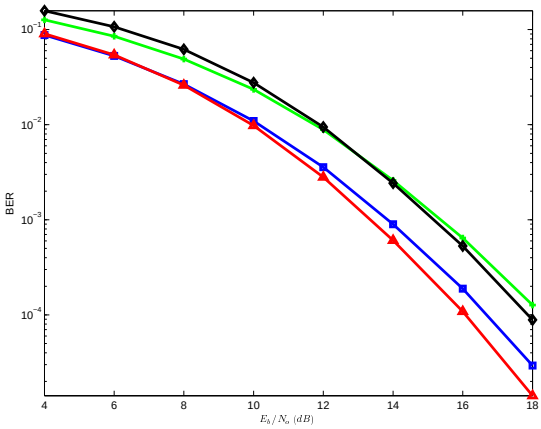
From Table 6.4, we see that quasiorthogonal and embedded orthogonal designs are favorable in terms of error rate performance compared to TAST and perfect codes. For example, for $q = 64$, the TAST code requires an additional 2.4 and 1.9 dB to achieve the



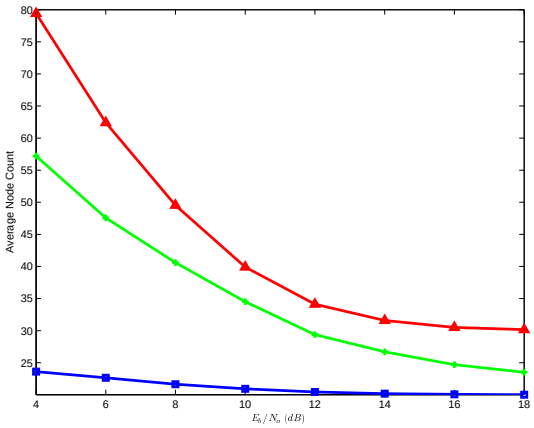
(a) Performance for $q = 4$



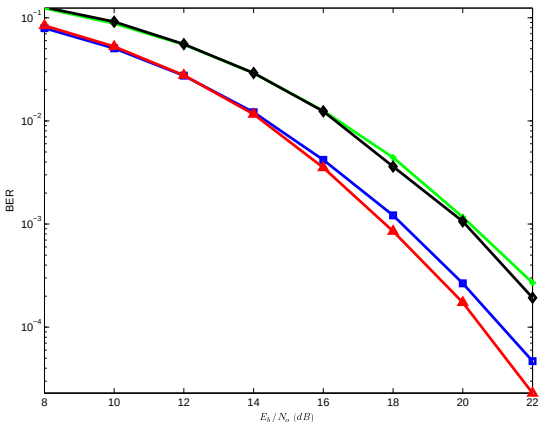
(b) Complexity for $q = 4$



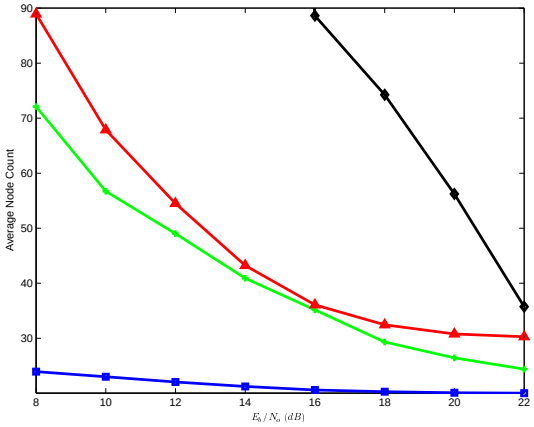
(c) Performance for $q = 16$



(d) Complexity for $q = 16$



(e) Performance for $q = 64$



(f) Complexity for $q = 64$

Figure 6.5: Performance and complexity comparison for $M = 6$, $\mathcal{R} = 1$ and $N = 1$ for quasiorthogonal ($\triangle$), TAST ($+$), perfect ($\diamond$) and embedded orthogonal space-time ($\square$) block codes.

same error rate as quasiorthogonal and embedded orthogonal codes, respectively. Unlike the case for $M = 4$, the TAST code has lower average complexity than the quasiorthogonal code. This is expected since the worst-case ML decoding complexity of the TAST code is $\mathcal{O}(q^{2.5})$, which is lower than that of the quasiorthogonal code with worst-case complexity of $\mathcal{O}(q^3)$. The embedded orthogonal code has the lowest decoding complexity. The embedded orthogonal code suffers a 0.5 dB performance penalty compared to the quasiorthogonal code, but it is at least 35% less complex than the quasiorthogonal code for any spectral efficiency. Furthermore, the worst-case ML complexity of the embedded orthogonal code is 93%, 99.5% and 99.9% lower than that of quasiorthogonal codes for $q = 4$, $q = 16$ and $q = 64$, respectively.

Table 6.4: Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 6$, $N = 1$, and $\mathcal{R} = 1$. Average Complexity and SNR Penalty are Relative to 9.3, 13.3 and 17.8 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively.

Code	Average Complexity			SNR Penalty (dB)		
	$q = 4$	$q = 16$	$q = 64$	$q = 4$	$q = 16$	$q = 64$
Quasiorthogonal	31.1	32.4	32.8	0	0	0
Embedded Orthogonal	20.1	20.2	20.3	0.50	0.50	0.50
Perfect	N/A	N/A	N/A	1.61	1.90	2.30
TAST	24.4	27.6	29.9	1.51	2.00	2.43

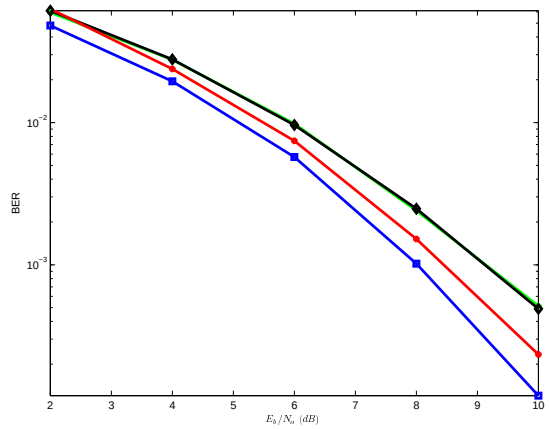
We have seen so far that the embedded orthogonal codes have the lowest decoding complexity compared to quasiorthogonal, TAST and perfect codes for $\mathcal{R} = 1$. Furthermore, the embedded orthogonal codes are second only to the quasiorthogonal designs and they are within a fraction of a dB in terms of the SNR required to achieve a target error probability. We next present results for $\mathcal{R} > 1$ and show that the embedded orthogonal designs are not only favorable in terms in decoding complexity, but they are also lower in error probability.

The complexity and performance results for $M = 4$, $N = 2$ and $\mathcal{R} = 1.5$ are shown in Figure 6.6. In addition to the perfect, TAST and embedded orthogonal space-time code, we show simulation results for a recently proposed code for the specific configuration of $M = 4$ and $\mathcal{R} = 1.5$ [66]. We only show simulation results for 4-QAM because the space-time

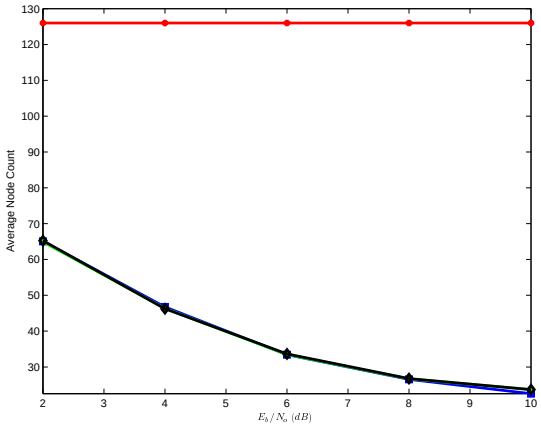
code was constructed by numerical optimization, and the authors note that optimizing the code for larger size alphabets became numerically infeasible. The worst-case ML decoding complexity for the code in [66] is $\mathcal{O}(q^3)$. A detection algorithm, which has the same fixed and worst-case decoding complexity of $\mathcal{O}(q^3)$ was also proposed in [66]. We can interpret the detection algorithm as a tree search, computing $\sqrt{q}^6$ path metrics in a six level real-valued tree. Hence, the number of nodes visited in the tree search is 2+4+8+16+32+64 = 126.

We see that the embedded orthogonal code has better performance than the perfect and TAST codes as well as the code in [66]. It is interesting, however, to note that the average complexity of embedded orthogonal, TAST and perfect codes is roughly the same, even though the worst-case complexity of the embedded orthogonal code is 83%, 97% and 99% lower than that of the perfect and TAST code. We rationalize that this is mainly due to the fact that the system is overdetermined such that there are more equations than unknowns. Specifically, the effective channel matrix is of size 16×12. Compared to a 12×12 effective channel matrix, the 16×12 matrix is less likely to have columns that can be expressed as a linear combination of the other columns. This results in an $\mathbf{R}$ matrix that has larger diagonal elements compared to the $\mathbf{R}$ matrix for 12×12 effective channel matrix. Having larger diagonal elements has two effects on the efficiency of the tree search. First, on average, the starting node at each layer is closer to the ML node. This results in reaching the ML solution in a fewer number of nodes visited. Second, the branch metric for an incorrect branch is larger for larger values of the diagonal elements. This results in pruning more tree branches since the sum of the branch metrics at a given level is more likely to exceed the sphere radius. Hence, we are less likely to spend our search effort in the upper tree branches. Given these two effects of having larger diagonal elements, it is not surprising that the average node count for the perfect, TAST and embedded orthogonal space-time codes is almost identical.

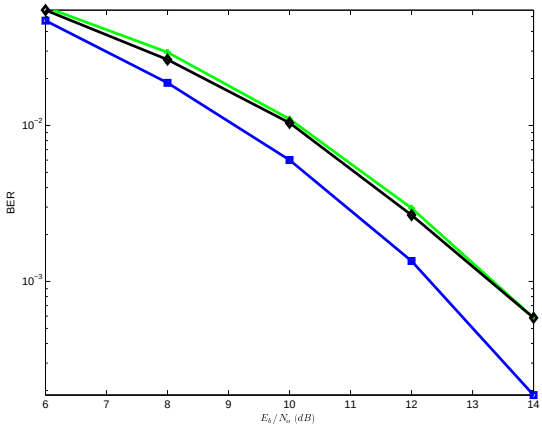
We summarize the complexity-performance results in Table 6.5. As can been seen from Table 6.5, the embedded orthogonal space-time block code is at least 1 dB better than the perfect and TAST codes, and it has the same average complexity. The worst-case ML



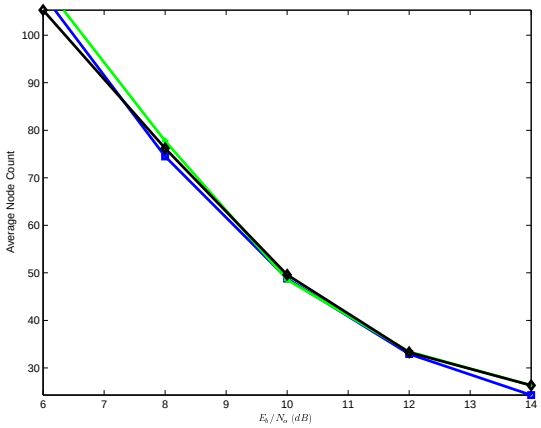
(a) Performance for $q = 4$



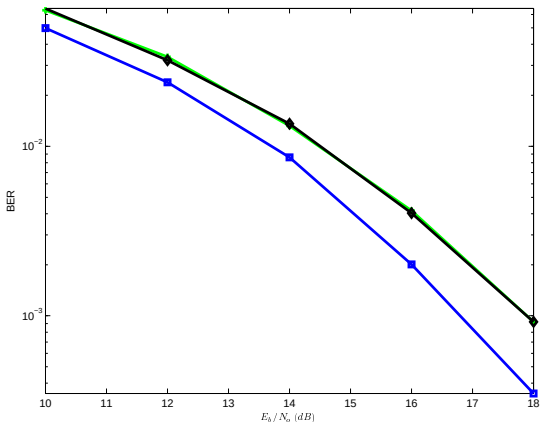
(b) Complexity for $q = 4$



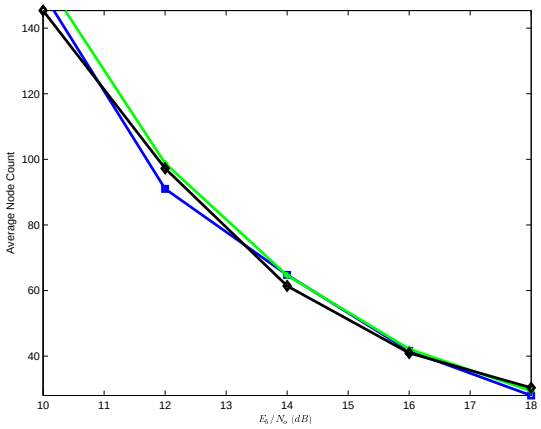
(c) Performance for $q = 16$



(d) Complexity for $q = 16$



(e) Performance for $q = 64$



(f) Complexity for $q = 64$

Figure 6.6: Performance and complexity comparison for $M = 4$, $\mathcal{R} = 1.5$ and $N = 2$ for TAST ($+$), perfect ($\diamond$), embedded orthogonal space-time ($\square$) block code and Ismail *et al.* space-time [66] ($*$) block code.

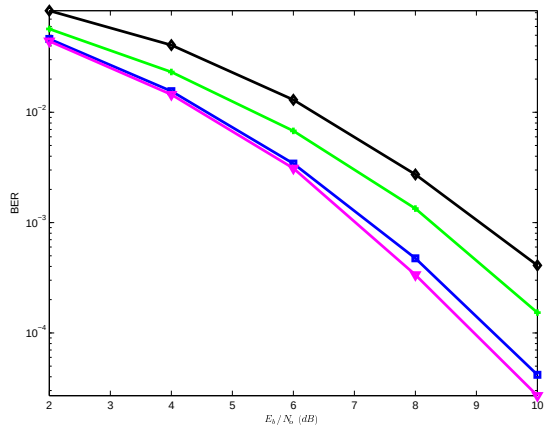
decoding complexity also favors the embedded orthogonal space-time block code over the TAST and perfect codes as mentioned earlier and shown in Table 6.2. Compared to the code in [66], the embedded orthogonal space-time block code is 0.4 dB better in terms of the SNR required to achieve 10^{-3} bit-error rate. It also has lower worst-case and lower average complexity. This is a rather interesting result given the fact the code in [66] is designed not only for low decoding complexity, but also for this one specific configuration and one specific alphabet size.

Table 6.5: Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 4$, $N = 2$, and $\mathcal{R} = 1.5$. Average Complexity and SNR Penalty are Relative to 8, 12.3 and 16.8 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively.

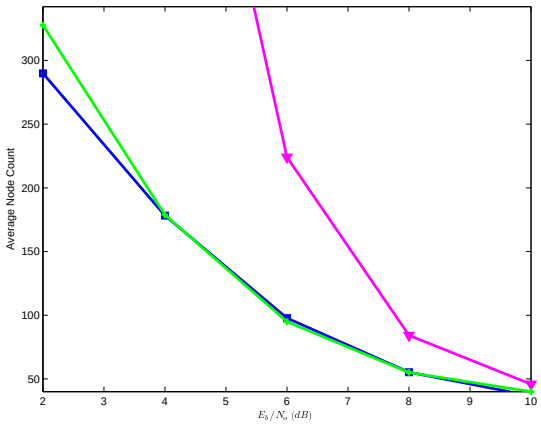
Code	Average Complexity			SNR Penalty (dB)		
	$q = 4$	$q = 16$	$q = 64$	$q = 4$	$q = 16$	$q = 64$
Embedded Orthogonal	26.5	31.6	36.2	0	0	0
Perfect	26.8	32.2	36.8	1.1	1.1	1.1
TAST	26.7	32.4	37	1.1	1.1	1.1
Ismail *et al.* [66]	126	N/A	N/A	0.4	N/A	N/A

We next discuss complexity and performance results for $M = 6$, $N = 2$ and $\mathcal{R} = 1.5$, which are shown in Figure 6.7. We see that the embedded $\boldsymbol{\mathcal{G}}(2, 1)$ and embedded $\boldsymbol{\mathcal{G}}(4, \frac{3}{4})$ orthogonal codes have better performance than the perfect and TAST codes. We omit complexity results for the perfect code since it is decoded using the complex-valued sphere decoder. We note, however, that even when we count a complex node as one real node, the perfect space-time code is more complex than the embedded and TAST codes for certain SNR values. For example, for $q = 64$ the perfect code is 40% and 70% more complex than the $\boldsymbol{\mathcal{G}}(4, \frac{3}{4})$ embedded code and TAST code, respectively, at the SNR required to achieve bit-error rate of 10^{-3}.

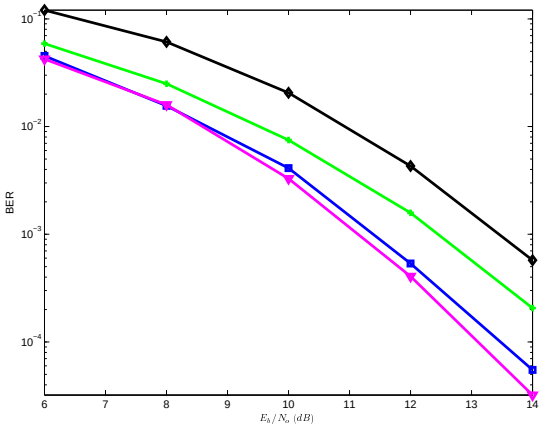
Similar to the case of $\mathcal{R} = 1.5$ for $M = 4$ antennas, we see that the TAST and $\boldsymbol{\mathcal{G}}(2, 1)$ embedded orthogonal code have roughly the same complexity. Another interesting result obtained in these simulation results is that the decoding complexity of the $\boldsymbol{\mathcal{G}}(4, \frac{3}{4})$ embedded orthogonal code is significantly higher than the TAST and $\boldsymbol{\mathcal{G}}(2, 1)$ embedded code. We



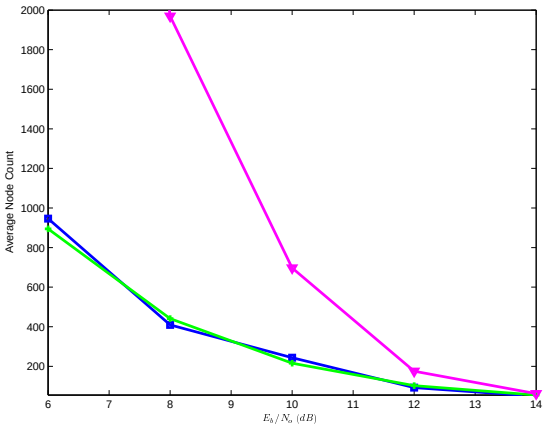
(a) Performance for $q = 4$



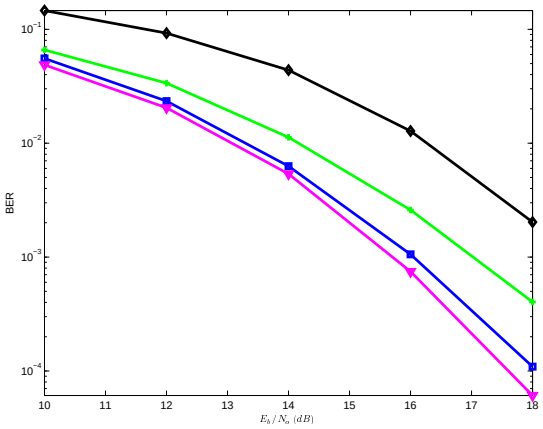
(b) Complexity for $q = 4$



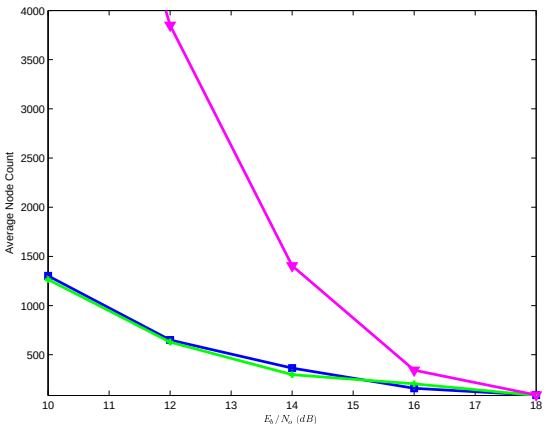
(c) Performance for $q = 16$



(d) Complexity for $q = 16$



(e) Performance for $q = 64$



(f) Complexity for $q = 64$

Figure 6.7: Performance and complexity comparison for $M = 6$, $\mathcal{R} = 1.5$ and $N = 2$ for TAST ($+$), perfect ($\diamond$), $\boldsymbol{\mathcal{G}}(2, 1)$ embedded orthogonal ($\square$) and $\boldsymbol{\mathcal{G}}(4, \frac{3}{4})$ embedded orthogonal ($\triangledown$) block codes.

rationalize that this due to the fact that there are more levels in the tree for the $\boldsymbol{\mathcal{G}}(4, \frac{3}{4})$ embedded code than for the other codes. Specifically, there are 24 levels in the tree search for the $\boldsymbol{\mathcal{G}}(4, \frac{3}{4})$ embedded code, compared to only 18 for the other codes. This is due to the fact that $\boldsymbol{\mathcal{G}}(4, \frac{3}{4})$ itself is a tall matrix of size 4×3. Hence, for the embedded code to achieve $\mathcal{R} = 1.5$, it transmits 12 complex information symbols in 8 signaling intervals. By comparison, the perfect, TAST and $\boldsymbol{\mathcal{G}}(2, 1)$ embedded code transmit only 9 complex information symbols in 6 signaling intervals. The complexity-performance tradeoff is summarized in Table 6.6.

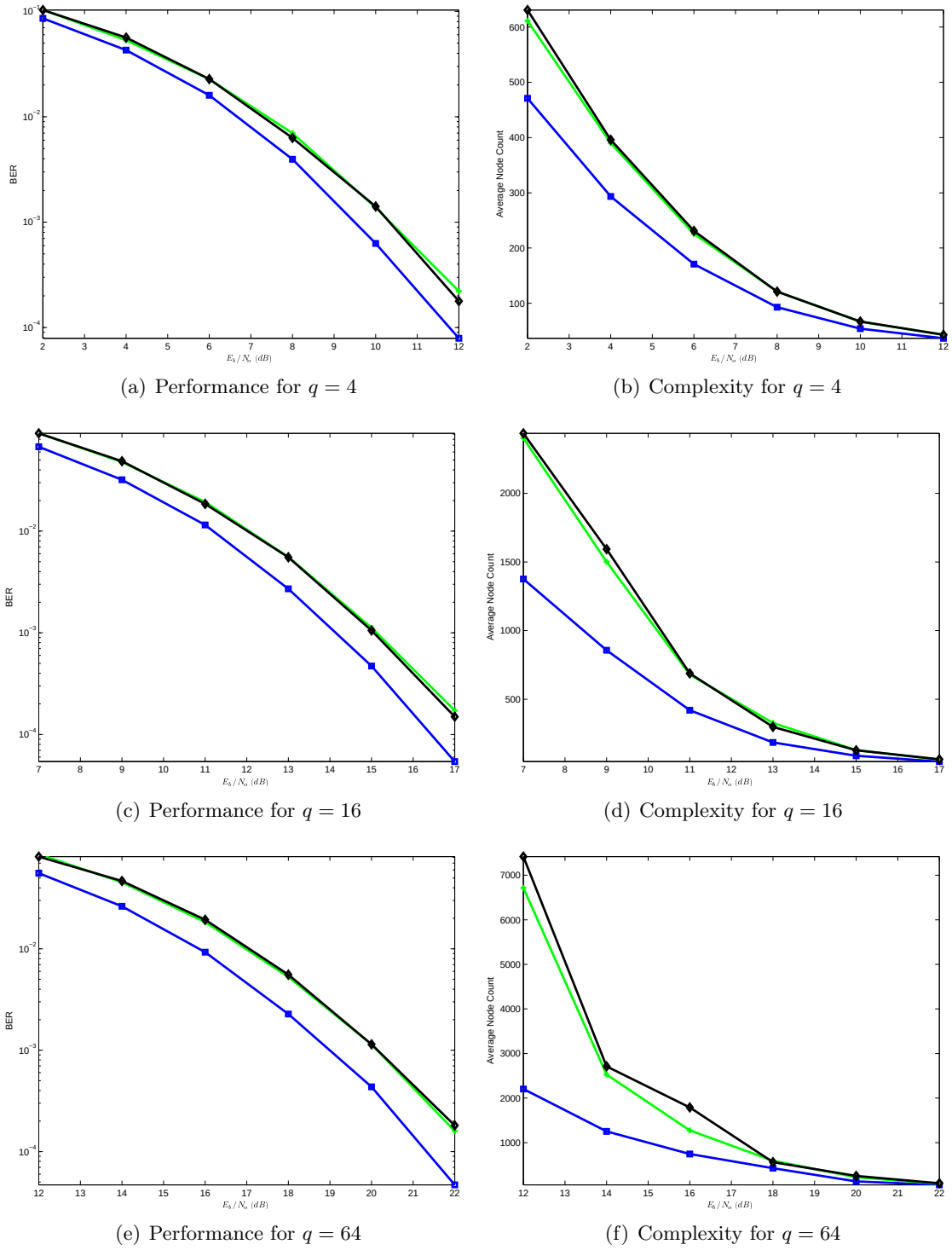
Table 6.6: Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 6$, $N = 2$, and $\mathcal{R} = 1.5$. Average Complexity and SNR Penalty are Relative to 7.1, 11.2 and 15.7 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively.

Code	Average Complexity			SNR Penalty (dB)		
	$q = 4$	$q = 16$	$q = 64$	$q = 4$	$q = 16$	$q = 64$
Embedded Orthogonal $\boldsymbol{\mathcal{G}}(4, \frac{3}{4})$	151	384	504	0	0	0
Embedded Orthogonal $\boldsymbol{\mathcal{G}}(2, 1)$	76	153	190	0.2	0.25	0.35
Perfect	N/A	N/A	N/A	2.05	2.25	2.45
TAST	74	147	210	1.25	1.25	1.3

Based on Table 6.6, we believe that the $\boldsymbol{\mathcal{G}}(2, 1)$ embedded orthogonal code offers the best complexity-performance tradeoff since the marginal increase in performance of the $\boldsymbol{\mathcal{G}}(4, \frac{3}{4})$ embedded code might not justify the significant increase in decoding complexity. The $\boldsymbol{\mathcal{G}}(2, 1)$ embedded orthogonal code is within 0.35 dB of the $\boldsymbol{\mathcal{G}}(4, \frac{3}{4})$ embedded code, and it is better performing than TAST and perfect codes by at least 1.25 and 2.05 dB, respectively.

Complexity and performance results for $M = 4$, $N = 2$ and $\mathcal{R} = 2$ are shown in Figure 6.8. We see that the embedded orthogonal space-time code is simultaneously lower in decoding complexity and better performing than the perfect and TAST codes. The complexity-performance tradeoff is summarized in Table 6.7.

As can be seen from Table 6.7, the embedded orthogonal space-time code is at least 0.85 dB better in terms of the SNR required to achieve error probability of 10^{-3} compared to TAST and perfect codes for any spectral efficiency. Furthermore, the embedded orthogonal



(a) Performance for $q = 4$

(b) Complexity for $q = 4$

(c) Performance for $q = 16$

(d) Complexity for $q = 16$

(e) Performance for $q = 64$

(f) Complexity for $q = 64$

Figure 6.8: Performance and complexity comparison for $M = 4$, $\mathcal{R} = 2$ and $N = 2$ for TAST (+), perfect ($\diamond$) and embedded orthogonal space-time ($\square$) codes.

Table 6.7: Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 4$, $N = 2$, and $\mathcal{R} = 2$. Average Complexity and SNR Penalty are Relative to 9.5, 14.2 and 19 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively.

Code	Average Complexity			SNR Penalty (dB)		
	$q = 4$	$q = 16$	$q = 64$	$q = 4$	$q = 16$	$q = 64$
Embedded Orthogonal	63	128	278	0	0	0
Perfect	82	197	407	0.85	0.88	1.17
TAST	81	208	408	0.85	0.92	1.16

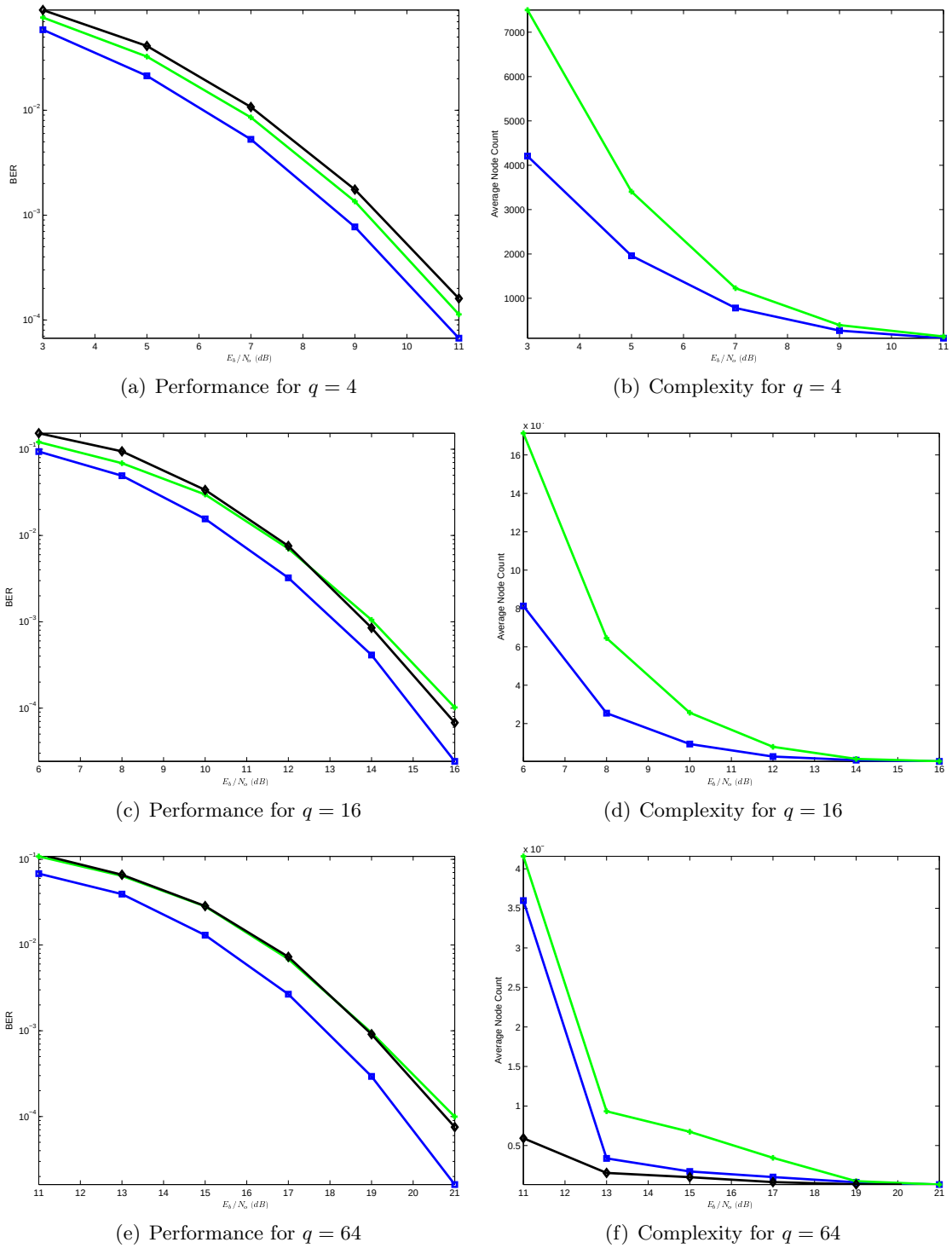
code is 22%, 35% and 31% less complex than the perfect and TAST codes for $q = 4$, $q = 16$ and $q = 64$, respectively.

Complexity and performance results for $M = 6$, $N = 2$ and $\mathcal{R} = 2$ are shown in Figure 6.9. We again omit complexity results for the perfect code since it is decoded using the complex-valued sphere decoder. We also note that even when we count a complex node as one real node, the perfect space-time code is more complex than the embedded and TAST codes for certain SNR values. Similar to the case of four transmit antennas, the embedded orthogonal space-time code is simultaneously lower in decoding complexity and better performing than the perfect and TAST codes. The complexity-performance tradeoff is summarized in Table 6.8.

Table 6.8: Average Complexity and Additional SNR Required to Achieve Bit-Error Probability of 10^{-3} for $M = 6$, $N = 2$, and $\mathcal{R} = 2$. Average Complexity and SNR Penalty are Relative to 8.7, 13.1 and 17.9 dB for $q = 4$, $q = 16$ and $q = 64$, Respectively.

Code	Average Complexity			SNR Penalty (dB)		
	$q = 4$	$q = 16$	$q = 64$	$q = 4$	$q = 16$	$q = 64$
Embedded Orthogonal	334	1750	7185	0	0	0
Perfect	N/A	N/A	N/A	0.70	0.70	1.00
TAST	497	4250	21200	0.50	0.90	1.05

As can be seen from Table 6.8, the embedded orthogonal space-time code is 0.5, 0.7 and 1 dB better than the perfect and TAST codes for $q = 4$, $q = 16$ and $q = 64$, respectively. Furthermore, the embedded orthogonal code is 33%, 59% and 66% less complex than the



(a) Performance for $q = 4$

(b) Complexity for $q = 4$

(c) Performance for $q = 16$

(d) Complexity for $q = 16$

(e) Performance for $q = 64$

(f) Complexity for $q = 64$

Figure 6.9: Performance and complexity comparison for $M = 6$, $\mathcal{R} = 2$ and $N = 2$ for TAST (+), perfect ($\diamond$) and embedded orthogonal space-time ($\square$) codes.

TAST code for $q = 4$, $q = 16$ and $q = 64$, respectively.

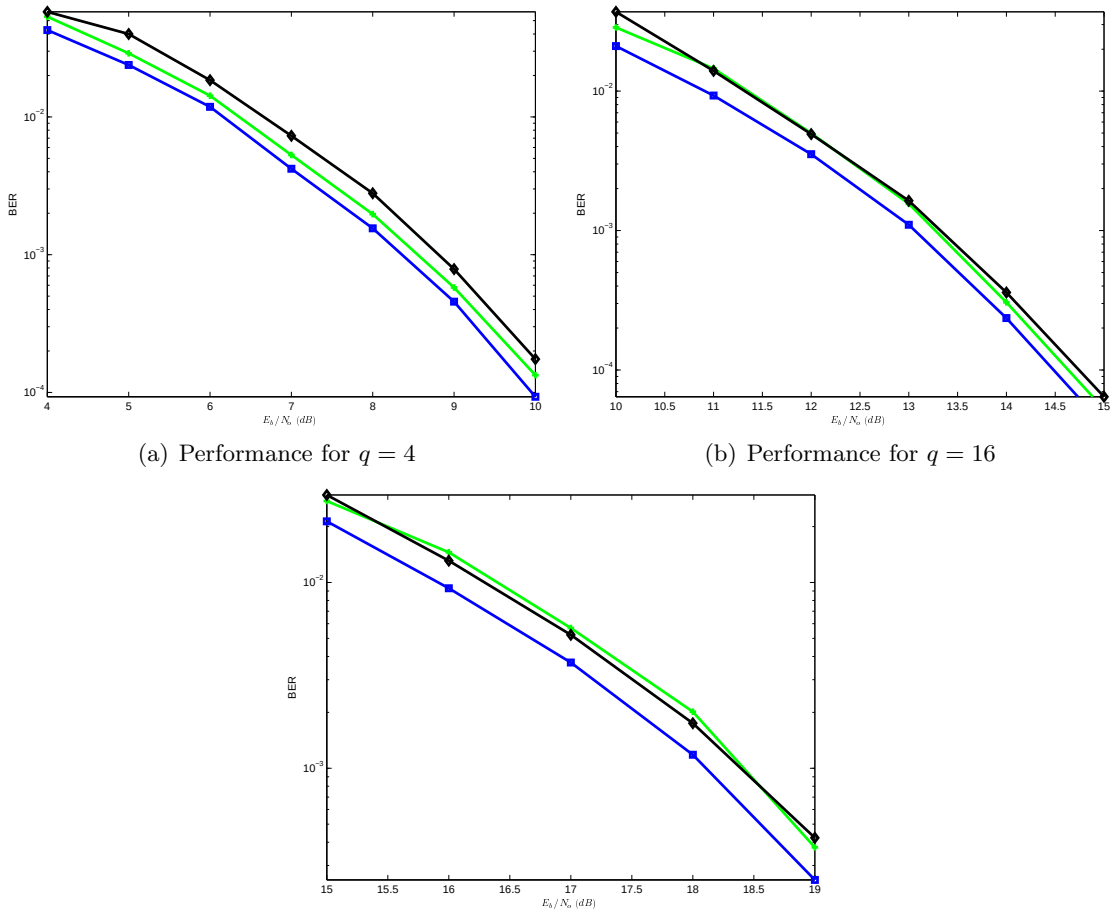
Performance results for $M = 6$, $N = 3$ and $\mathcal{R} = 3$ are shown in Figure 6.10. We omit complexity comparison because the results obtained are statistically insignificant due to the small number of channel realizations used to obtain the results. For example, 2000 channel realizations at 4 dB SNR for 4-QAM generated more than 2000 bit errors for the TAST curve (and other curves as well). However, it was observed that for one channel realization, the sphere decoder visited more than 2.5 billion nodes, while for the rest of the channel realizations, the sphere decoder visited less than 10,000 nodes, on average. By including that one particular realization with more than 2.5 billion nodes visited in the average node count, the average node count jumps two orders of magnitude to more than 1 million nodes.

As can be seen from Figure 6.10, the embedded orthogonal space-time block code is more than 0.2 dB better than the perfect and TAST code in terms of the SNR required to achieve error probability of 10^{-3} for any spectral efficiency.

We have shown through simulations that the embedded orthogonal space-time block codes have lower average decoding complexity than the quasiorthogonal, perfect and TAST codes for up to six transmit antennas. This result is not surprising given the fact the embedded orthogonal codes have the lowest worst-case decoding complexity compared to the quasiorthogonal, perfect and TAST codes as shown in Table 6.2. Furthermore, simulation results show that the error rate performance of the embedded orthogonal codes is better than the perfect and TAST codes for any rate and any spectral efficiency up to six transmit antennas and bit-error rate of 10^{-4} or higher.

6.4 Conclusions

We have proposed a family of space-time codes for an arbitrary number of transmit antennas and rates up to half the number of transmit antennas. We introduced the concept of embedded orthogonal space-time codes, in which complex orthogonal designs assume the role of complex information symbols in the encoding process. The proposed family of embedded orthogonal codes has the lowest decoding complexity for any rate up to half the number of transmit antennas. Furthermore, simulation results up to six transmit antennas show that



(a) Performance for $q = 4$

(b) Performance for $q = 16$

(c) Performance for $q = 64$

Figure 6.10: Performance comparison for $M = 6$, $\mathcal{R} = 3$ and $N = 3$ for TAST (+), perfect ($\diamond$) and embedded orthogonal space-time ($\square$) block codes.

the proposed embedded orthogonal space-time codes have better error rate performance than previous constructions for all rates higher than one symbol per signaling interval. For transmission rate of one symbol per signaling interval, the embedded orthogonal space-time codes is within a fraction of a dB of the best performing space-time code, but it requires lower decoding complexity.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this thesis, we constructed space-time block codes with the aim of low complexity detection at the receiver. Furthermore, we developed efficient maximum-likelihood decoding algorithms of space-time block codes, with low average complexity. We next summarize the contributions of this thesis to the transmitter and receiver design, and we highlight some potential areas for future research.

7.1 Contributions

Because the worst-case decoding complexity plays a central theme in this thesis, we presented a unified framework for comparing space-time block codes in terms of their worst-case decoding complexity. Using this framework, we showed that certain families of space-time block codes have lower decoding complexity than an exhaustive-search decoder. One dramatic example that shows the value of our framework is the case of the *golden code*.

The golden code is one of the most important constructions of space-time block codes for the two-input two-output channel offering not only full diversity but also full rate of two symbols per signaling interval. Furthermore, the golden code achieves the diversity-multiplexing tradeoff and in terms of the signal-to-noise ratio required to achieve a target error probability, it performs better than all previously reported full-rate codes with two transmit antennas. An exhaustive-search decoder for the golden code has a worst-case decoding complexity grows with the fourth power of the alphabet size. We proved, however, that the worst-case decoding complexity is sub-cubic in the alphabet size. Specifically, we proved that the worst-case decoding complexity is $\mathcal{O}\left(q^{2.5}\right)$, where q is the alphabet size. Furthermore, we proposed an efficient maximum-likelihood decoder that is not only 30% lower in average complexity than a conventional sphere detector, but it outperforms a conventional detector in terms of the worst-case decoding complexity, sort complexity and decoding latency.

Device mobility results in channels that could vary quickly between quasistatic fading and rapid time-varying fading, depending on the mobile speed or Doppler frequency. Therefore, space-time block codes with reduced decoding complexity in both quasistatic and rapid time-varying fading channels are of great practical importance. To this end, we revisited the practically important two-transmit and two-receive antenna system and proposed the *asymmetric golden code*. The asymmetric golden code is optimal in terms of the diversity-multiplexing tradeoff, and it has a lower worst-case decoding complexity than the golden code. Specifically, the worst-case decoding complexity is $\mathcal{O}\left(q^2\right)$. Compared to previous constructions for the two-input two-output channel, the asymmetric golden code has the lowest decoding complexity and maintains its low decoding complexity on both quasistatic and rapid time-varying channels. Furthermore, we proposed and efficient maximum-likelihood decoder with low average complexity. Simulation results show that the asymmetric golden code is an attractive alternative to other constructions in terms of its complexity-performance tradeoff.

The desire for achieving higher data rates provided the motivation to construct high-rate space-time block codes, whose transmission rate is higher than one symbol per signaling interval. Two important families were constructed that achieve full-rate equal to the number of transmit antennas, these are the perfect and threaded algebraic space-time block codes. A drawback to the perfect and threaded algebraic codes is that they encode only the information symbols, but not their conjugates. This results in an effective channel matrix that contains only the fading coefficient, but not their conjugates, and hence, the effective channel matrix cannot have orthogonal columns, in general. Therefore, unlike orthogonal, quasiorthogonal and semiorthogonal algebraic space-time codes, the perfect and threaded algebraic space-time block codes cannot exploit any reduction in decoding complexity that would be possible due to the orthogonality among the columns of the effective channel matrix.

We overcome the drawback of encoding only the information symbols in high-rate space-time codes by introducing the concept of *embedding*, in which orthogonal designs subsume the role of information symbols in the encoding process. The embedding concept was then

used to construct the *embedded orthogonal space-time block codes*. Because information symbols appear along with their conjugates in the space-time code matrix, fading coefficients along with their conjugates also appear in the effective channel matrix, inducing orthogonality among its columns. The embedded orthogonal space-time codes subsume two important families as special cases, the single-symbol decodable space-time codes and the semiorthogonal algebraic space-time block codes. When compared to previously reported space-time codes including quasiorthogonal, threaded algebraic and perfect space-time block codes with the same number of antennas and the same rate, the embedded orthogonal space-time codes have the lowest worst-case decoding complexity. Furthermore, simulation result show that the embedded orthogonal codes are also lower in error probability on quasistatic Rayleigh channels for up to six transmit antennas and rates higher than one symbol per signaling interval.

7.2 Future Work

The space-time block codes discussed in this research are not separable for rates higher than one symbol per signaling interval. In other words, the decoding cannot be done over two or more independent groups of symbols. One important contribution in this area would be to establish necessary conditions for the existence of full diversity and high-rate separable space-time codes. Equally as important is the determination of the maximum achievable rate of a fully diverse space-time block code for a given number of antennas.

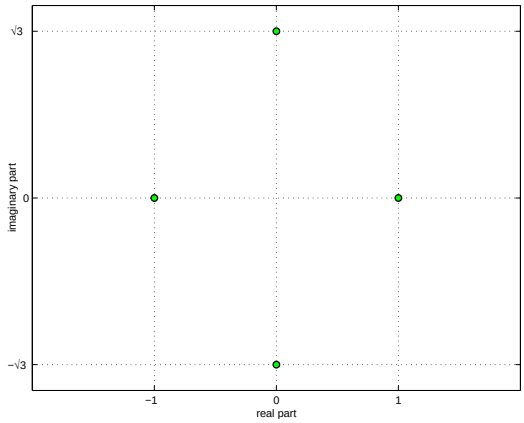
To our knowledge, the first separable high-rate design reported in literature is due to Yuen-Guan-Tjhung [67]. In particular, a rate $\frac{5}{4}$ design was proposed for four transmit antennas. The proposed design in [67] not only sacrificed the transmission rate, but it also sacrificed the transmission diversity such that it only achieves a second order diversity. Another construction of rate $\frac{5}{4}$ space-time code was proposed for four transmit antennas in [68]. The proposed code in [68] also has second-order transmit diversity and as a result, orthogonal and quasiorthogonal space-time block codes outperform the code in [68] at high signal-to-noise ratio values. The maximum transmission rate of separable space-time block codes, as well the systematic design of such codes remain open research problems.

Another interesting direction for future research is the development of space-time block codes that achieve full diversity when low complexity suboptimal decoders are used. Several computationally efficient suboptimal decoders are proposed in the literature. These suboptimal decoders vary in complexity, with some requiring only linear complexity, but they do not achieve the optimal performance of the ML decoder. Hence, they offer a trade-off between complexity and performance. Suboptimal decoders include the zero-forcing and minimum mean-squared error linear equalizers, the decision-feedback equalizer, the lattice-reduction-aided equalizer [69], and the fixed-complexity variants of the sphere decoding algorithm [70][71]. The interested reader is referred to [72] for an extensive review of suboptimal MIMO decoders.

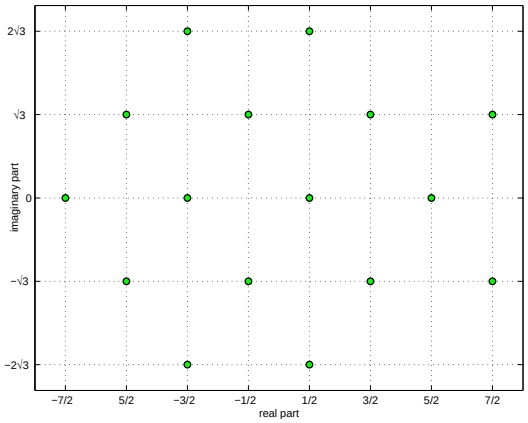
A recent contribution in this direction is the development of space-time block codes that achieve full diversity when a linear equalizer is used to separate the decoding of the transmitted symbols into two or more groups [73]. This approach is different from the separable approach to the design of space-time block codes in the fact that the space-time block codes developed in [73] are not separable until *after* a linear equalizer is used to separate the decoding into two or more groups. Full transmit diversity can then be harnessed at the receiver using an ML decoder applied to each group. The maximum rate of these space-time block codes is $\frac{4}{3}$ for two antennas, $\frac{8}{5}$ for four antennas, and upper bounded by 2 for larger number of antennas.

APPENDIX A

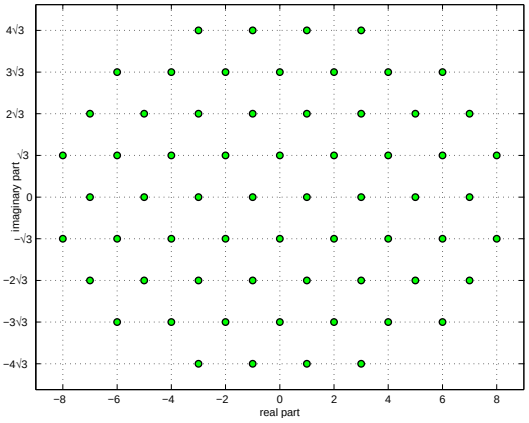
OPTIMUM 4, 16 AND 64-POINT HEXAGONAL CONSTELLATIONS



(a) $q = 4$



(b) $q = 16$



(c) $q = 64$

Figure A.1: Optimum hexagonal constellations for $q = 4$, $q = 16$ and $q = 64$.

APPENDIX B

PROOF OF THE KEY PROPERTY FOR FAST DECODING OF THE GOLDEN CODE

We will use a QR decomposition of $\bar{\mathbf{H}}$ from (4.4), namely $\bar{\mathbf{H}} = \bar{\mathbf{Q}}\bar{\mathbf{R}}$, to construct a QR decomposition of $\mathbf{H} = \bar{\mathbf{H}}\Psi$, namely $\mathbf{H} = \mathbf{QR}$.

Inspection of (4.4) reveals that $\bar{\mathbf{h}}_1^*\mathbf{h}_2 = \bar{\mathbf{h}}_1^*\mathbf{h}_4 = \bar{\mathbf{h}}_2^*\mathbf{h}_3 = \bar{\mathbf{h}}_3^*\mathbf{h}_4 = 0$, which implies that the subspace spanned by the first and third columns of $\bar{\mathbf{H}}$ is orthogonal to the subspace spanned by the second and fourth columns. This fact implies that $\bar{r}_{1,2} = \bar{r}_{1,4} = \bar{r}_{2,3} = \bar{r}_{3,4} = 0$, so that:

$$\begin{aligned}
\mathbf{H} &= \bar{\mathbf{H}}\Psi \\
&= \bar{\mathbf{Q}} \begin{bmatrix} \bar{r}_{1,1} & 0 & \bar{r}_{1,3} & 0 \\ 0 & \bar{r}_{2,2} & 0 & \bar{r}_{2,4} \\ 0 & 0 & \bar{r}_{3,3} & 0 \\ 0 & 0 & 0 & \bar{r}_{4,4} \end{bmatrix} \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \\
&= \bar{\mathbf{Q}}\mathbf{F}, \qquad \text{(B.1)}
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{F} &= \begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{0} & \mathbf{Z} \end{bmatrix}, \\
\mathbf{X} &= \begin{bmatrix} c\bar{r}_{1,1} & s\bar{r}_{1,1} \\ -s\bar{r}_{2,2} & c\bar{r}_{2,2} \end{bmatrix}, \\
\mathbf{Y} &= \begin{bmatrix} c\bar{r}_{1,3} & s\bar{r}_{1,3} \\ -s\bar{r}_{2,4} & c\bar{r}_{2,4} \end{bmatrix}, \\
\mathbf{Z} &= \begin{bmatrix} c\bar{r}_{3,3} & s\bar{r}_{3,3} \\ -s\bar{r}_{4,4} & c\bar{r}_{4,4} \end{bmatrix}. \qquad \text{(B.2)}
\end{aligned}$$

Observe that the submatrices $\mathbf{X}$ and $\mathbf{Z}$ are entirely real, since c and s and $\bar{r}_{ii}$, $i \in \{1, 2, 3, 4\}$,

are all real. Therefore, we can transform $\mathbf{F}$ into an upper triangular matrix $\mathbf{R} = \mathbf{WF}$ via the purely real Givens rotation matrix:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2 \end{bmatrix} \tag{B.3}$$

where

$$\begin{aligned} \mathbf{W}_1 &= \frac{1}{\sqrt{(c\bar{r}_{1,1})^2 + (s\bar{r}_{2,2})^2}} \begin{bmatrix} c\bar{r}_{1,1} & -s\bar{r}_{2,2} \\ s\bar{r}_{2,2} & c\bar{r}_{1,1} \end{bmatrix}, \\ \mathbf{W}_2 &= \frac{1}{\sqrt{(c\bar{r}_{3,3})^2 + (s\bar{r}_{4,4})^2}} \begin{bmatrix} c\bar{r}_{3,3} & -s\bar{r}_{4,4} \\ s\bar{r}_{4,4} & c\bar{r}_{3,3} \end{bmatrix}. \end{aligned} \tag{B.4}$$

Substituting $\mathbf{F} = \mathbf{W}^\top \mathbf{R}$ into (B.1) yields the desired QR decomposition $\mathbf{H} = \mathbf{QR}$, where $\mathbf{Q} = \bar{\mathbf{Q}}\mathbf{W}^\top$ and

$$\mathbf{R} = \mathbf{WF} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2 \end{bmatrix} \begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{0} & \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}. \tag{B.5}$$

Since $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{X}$ and $\mathbf{Z}$ are all real, it follows that both $\mathbf{A} = \mathbf{W}_1\mathbf{X}$ and $\mathbf{D} = \mathbf{W}_2\mathbf{Z}$ are real. And by construction of $\mathbf{W}_1$ and $\mathbf{W}_2$, both $\mathbf{A}$ and $\mathbf{D}$ are upper triangular. □

APPENDIX C

OPTIMAL ASYMMETRY COEFFICIENT $\mathcal{K}$

For the asymmetric golden code, the coding gain in (2.12) can be written as

$$\begin{aligned}\Gamma &= \min_{\mathbf{C}-\tilde{\mathbf{C}}} \left(\det(\mathbf{C}-\tilde{\mathbf{C}})^*(\mathbf{C}-\tilde{\mathbf{C}})\right)^{1/M} \\ &= \min_{\mathbf{C}-\tilde{\mathbf{C}}} \left(|\det(\mathbf{C}-\tilde{\mathbf{C}})|^2\right)^{1/2} \\ &= \min_{\mathbf{C}-\tilde{\mathbf{C}}} |\det(\mathbf{C}-\tilde{\mathbf{C}})|. \end{aligned} \tag{C.1}$$

Since we are maximizing the coding gain for QAM alphabet, which is a subset of $\mathbb{Z}[i] \triangleq \{a+bi\}$, $a, b \in \mathbb{Z}$, where $i = \sqrt{-1}$, and because the asymmetric golden code is linear, the asymptotic coding gain of (C.1) can be written as:

$$\Gamma(\mathcal{K}) = \min_{(\Delta x_1, \Delta x_2, \Delta x_3, \Delta x_4) \in \mathcal{M}} |\det(\mathbf{C})|, \tag{C.2}$$

where $\mathcal{M} \triangleq \{\mathbb{Z}+i\mathbb{Z}\}^4 - \{0,0,0,0\}$ is the 4-tuple of all possible complex integers excluding $\{\Delta x_1, \Delta x_2, \Delta x_3, \Delta x_4\} = \{0,0,0,0\}$, which is the all zero 4-tuple. Since the difference between two integers is again an integer, we simplify the coding gain expression as follows

$$\Gamma(\mathcal{K}) = \min_{(x_1, x_2, x_3, x_4) \in \mathcal{M}} |\det(\mathbf{C})|. \tag{C.3}$$

Substituting the definition of $\mathbf{C}$ in (5.13) into (C.3) yields:

$$\begin{aligned}\Gamma(\mathcal{K}) &= \min_{(x_1, x_2, x_3, x_4) \in \mathcal{M}} \left| \frac{2}{1+\mathcal{K}^2}(u_1 u_2^* - \mathcal{K}^2 u_3 u_4^*) \right| \\ &= \frac{1}{\sqrt{5}} \min_{(x_1, x_2, x_3, x_4) \in \mathcal{M}} \left| \frac{2}{1+\mathcal{K}^2} \left(g(x_1, x_2) - \mathcal{K}^2 g(x_3, x_4)\right) \right|, \end{aligned} \tag{C.4}$$

where

$$g(a,b) = \left(-|a|^2 + |b|^2 + \Re\{a^* b\} - i \cdot \sqrt{5}\Im\{a^* b\}\right). \tag{C.5}$$

We next find the value of $\mathcal{K}$ that maximizes (C.4). We proceed in two steps: first, we establish the bound $\Gamma(\mathcal{K}) \leq 1/\sqrt{20}$; then, we show that $\mathcal{K} = 1/\sqrt{3}$ achieves the bound with equality.

To establish the bound, let us introduce $\mathcal{L} \subseteq \{(0,0,1,0),(1,0,1,0),(1,0,1+i,0)\} \subseteq \mathcal{M}$. Because $\mathcal{L}$ is a subset of $\mathcal{M}$, it clearly follows that $\min_{\mathcal{M}}\{\,\cdot\,\} \leq \min_{\mathcal{L}}\{\,\cdot\,\}$, so that the coding gain of (C.4) can be bounded by:

$$\begin{aligned}
\Gamma(\mathcal{K}) &\leq \frac{1}{\sqrt{5}} \min_{(x_1,x_2,x_3,x_4)\in\mathcal{L}} \left| \frac{2}{1+\mathcal{K}^2}\left(g(x_1,x_2) - \mathcal{K}^2 g(x_3,x_4)\right)\right| \\
&= \frac{1}{\sqrt{5}} \min\left\{ \frac{2}{1+\mathcal{K}^2}\mathcal{K}^2, \frac{2}{1+\mathcal{K}^2}\left(1-\mathcal{K}^2\right), \frac{2}{1+\mathcal{K}^2}\left|1-2\mathcal{K}^2\right| \right\} \\
&\leq \frac{2}{\sqrt{5}} \max_{1>\mathcal{K}>0} \min\left\{ \frac{\mathcal{K}^2}{1+\mathcal{K}^2}, \frac{1-\mathcal{K}^2}{1+\mathcal{K}^2}, \frac{\left|1-2\mathcal{K}^2\right|}{1+\mathcal{K}^2} \right\} \qquad \text{(C.6)} \\
&= \frac{1}{2\sqrt{5}} = \frac{1}{\sqrt{20}}. \qquad \text{(C.7)}
\end{aligned}$$

The inequality of (C.6) follows from the fact that $h(\mathcal{K}) \leq \max_{1>\mathcal{K}>0}\{h(\mathcal{K})\}$ for any function $h(\mathcal{K})$ and for any $\mathcal{K} > 0$. The equality in (C.7) is a result from the following lemma.

Lemma C.1. *The maximum of the minimum is:*

$$\max_{1>\mathcal{K}>0} \min\left\{ \frac{\mathcal{K}^2}{1+\mathcal{K}^2}, \frac{1-\mathcal{K}^2}{1+\mathcal{K}^2}, \frac{\left|1-2\mathcal{K}^2\right|}{1+\mathcal{K}^2} \right\} = \frac{1}{4}, \qquad \text{(C.8)}$$

and is achieved for $\mathcal{K} = 1/\sqrt{3}$.

Proof. The functions $\mathcal{K}^2/(1+\mathcal{K}^2)$ and $(1-\mathcal{K}^2)/(1+\mathcal{K}^2)$ are monotonically increasing and decreasing, respectively. The function $|1-2\mathcal{K}^2|/(1+\mathcal{K}^2)$ is decreasing for $\mathcal{K} \in \left(0, 1/\sqrt{2}\,\right]$ and increasing for $\mathcal{K} \in (1/\sqrt{2}, 1)$. We find the maximum of the minimum over two regions; $\mathcal{K} \in \left(0, 1/\sqrt{2}\,\right]$ and $\mathcal{K} \in (1/\sqrt{2}, 1)$.

For $\mathcal{K} \in \left(0, 1/\sqrt{2}\,\right]$, the maximum of the minimum of (C.8) occurs at the intersection of the increasing function $\mathcal{K}^2/(1+\mathcal{K}^2)$ and the smaller of the two decreasing functions $(1-\mathcal{K}^2)/(1+\mathcal{K}^2)$ and $|1-2\mathcal{K}^2|/(1+\mathcal{K}^2)$. Since $|1-2\mathcal{K}^2| < (1-\mathcal{K}^2)$ for $\mathcal{K} \in \left(0, 1/\sqrt{2}\,\right]$, the maximum of the minimum occurs at the intersection of $\mathcal{K}^2/(1+\mathcal{K}^2)$ and $|1-2\mathcal{K}^2|/(1+\mathcal{K}^2)$, namely at $\mathcal{K} = 1/\sqrt{3}$. The maximum of the minimum is given by $\frac{1}{4}$.

For $\mathcal{K} \in (1/\sqrt{2}, 1)$, the maximum of the minimum occurs at the intersection of the decreasing function $(1-\mathcal{K}^2)/(1+\mathcal{K}^2)$ and the smaller of the two increasing functions $\mathcal{K}^2/(1+\mathcal{K}^2)$ and $|1-2\mathcal{K}^2|/(1+\mathcal{K}^2)$. Since $|1-2\mathcal{K}^2| < \mathcal{K}^2$ for $\mathcal{K} \in (1/\sqrt{2}, 1)$, the maximum of the minimum occurs at the intersection of $|1-2\mathcal{K}^2|$ and $(1-\mathcal{K}^2)$, namely at $\mathcal{K} = \sqrt{2/3}$.

The maximum of the minimum is given by 1/5 . Therefore, the maximum of the minimum is $\frac{1}{4}$ and occurs at $\mathcal{K} = 1/\sqrt{3}$. □

We next show that the bound $\Gamma(\mathcal{K}) \leq \frac{1}{2\sqrt{5}}$ of (C.7) is in fact achievable with equality with $\mathcal{K} = 1/\sqrt{3}$. Substituting $\mathcal{K} = 1/\sqrt{3}$ into (C.4), we have:

$$\Gamma(1/\sqrt{3}) = \frac{1}{2\sqrt{5}} \min_{(x_1,x_2,x_3,x_4)\in\mathcal{M}} |3g(x_1,x_2) - g(x_3,x_4)| . \tag{C.9}$$

In order to show that $\Gamma(1/\sqrt{3}) = \frac{1}{2\sqrt{5}}$, we first prove that $\Gamma(1/\sqrt{3}) = 0$ only if $x_1 = x_2 = x_3 = x_4 = 0$ in the following lemma.

Lemma C.2. *The function* $f(x_1, x_2, x_3, x_4) = 3g(x_1, x_2) - g(x_3, x_4) = 0$, *for* $x_i \in \{\mathbb{Z} + i\mathbb{Z}\}$, $i \in \{1, 2, 3, 4\}$, *only if* $x_1 = x_2 = x_3 = x_4 = 0$.

Proof. Assume that $f(x_1, x_2, x_3, x_4) = 0$, then we have that

$$3g(x_1, x_2) - g(x_3, x_4) = 0. \tag{C.10}$$

The equation in (C.10) implies that 3 divides $g(x_3, x_4)$, which is denoted $3|g(x_3, x_4)$. We next show that if $3|g(a, b)$, then $3|a$, $3|b$ and $9|g(a, b)$.

Let $a = 3l_a + r_a$ and $b = 3l_b + r_b$, where $l_a, l_b \in \{\mathbb{Z} + i\mathbb{Z}\}$, and $r_a, r_b \in \{0, 1, 2\} + i\{0, 1, 2\}$. Since, $3|g(a, b)$, we have that $3|g(r_a, r_b)$. By considering the finite number of possibilities for r_a and r_b, we can easily verify that $3|g(r_a, r_b)$ only if $r_a = r_b = 0$. Therefore, $3|a$ and $3|b$. Consequently, $g(a, b)$ is then given by

$$\begin{aligned} g(a,b) &= \left(-|3l_a|^2 + |3l_b|^2 + \Re\{3l_a^* 3l_b\} - i \cdot \sqrt{5}\Im\{3l_a^* 3l_b\}\right) \\ &= 9\left(-|l_a|^2 + |l_b|^2 + \Re\{l_a^* l_b\} - i \cdot \sqrt{5}\Im\{l_a^* l_b\}\right). \end{aligned} \tag{C.11}$$

As a result, we also have that $9|g(a, b)$.

Therefore, $f(x_1, x_2, x_3, x_4) = 0$ implies that $3|g(x_3, x_4)$, which in turn implies that $3|x_3$, $3|x_4$, and $9|g(x_3, x_4)$. However, $9|g(x_3, x_4)$ implies that $3|g(x_1, x_2)$, which in turn implies that $3|x_1$, $3|x_2$, and $9|g(x_1, x_2)$. We then have that $3|x_k$, $k \in \{1, 2, 3, 4\}$. Hence, we can divide (C.10) by 3 and obtain an identical equation in the coefficients $y_k = \frac{x_k}{3}$, where $y_k \in \{\mathbb{Z} + i\mathbb{Z}\}$. Since we can repeat this argument and divide by 3 indefinitely, the only possible solution to $f(x_1, x_2, x_3, x_4) = 0$ is when $x_1 = x_2 = x_3 = x_4 = 0$. □

The desired result that $\Gamma(1/\sqrt{3}) = \frac{1}{2\sqrt{5}}$ follows from (C.9) and Lemma C.2. In particular, Lemma C.2 proves that $|3g(x_1, x_2) - g(x_3, x_4)| \neq 0$ for $(x_1, x_2, x_3, x_4) \in \mathcal{M}$. Since the real part of $3g(x_1, x_2) - g(x_3, x_4)$ is an integer, while the imaginary part is an integer multiple of $\sqrt{5}$, it follows immediately that:

$$\min_{(x_1,x_2,x_3,x_4)\in\mathcal{M}} |3g(x_1, x_2) - g(x_3, x_4)| = 1. \tag{C.12}$$

Substituting (C.12) into (C.9), we obtain that $\Gamma(1/\sqrt{3}) = \frac{1}{2\sqrt{5}}$. □

APPENDIX D

PROOF OF THE KEY PROPERTY FOR FAST DECODING OF THE ASYMMETRIC GOLDEN CODE

Direct computation of the elements of $\mathbf{R}$ from the effective channel matrix in (5.21) yields:

$$\begin{aligned}
r_{1,1} &= \sqrt{\frac{3}{2}}\sqrt{c^2\mu_1 + s^2\mu_2} \\
r_{1,2} &= 0 \\
r_{1,3} &= \frac{3}{2\sqrt{5}r_{1,1}}(\mu_1 - \mu_2) \\
r_{1,4} &= \frac{\sqrt{3}}{2r_{1,1}}\mu_3 \\
r_{2,2} &= \frac{1}{\sqrt{3}}\, r_{1,1} \\
r_{2,3} &= \sqrt{3}\, r_{1,4}^* \\
r_{2,4} &= \frac{-1}{\sqrt{3}}\, r_{1,3} \\
r_{3,3} &= \frac{\sqrt{27}|\mu_4|}{\sqrt{8}r_{1,1}^2}\sqrt{c^2\mu_1 + s^2\mu_2} \\
r_{3,4} &= 0 \\
r_{4,4} &= \frac{1}{\sqrt{3}}\, r_{3,3},
\end{aligned} \tag{D.1}$$

where these results are expressed in terms of the following four intermediate variables:

$$\begin{aligned}
\mu_1 &= |h_{1,1}|^2 + |h_{1,2}|^2 \\
\mu_2 &= |h_{2,1}|^2 + |h_{2,2}|^2 \\
\mu_3 &= h_{1,1}^* h_{2,1} + h_{1,2}^* h_{2,2} \\
\mu_4 &= h_{1,1}h_{2,2} - h_{1,2}h_{2,1}.
\end{aligned} \tag{D.2}$$

Because fading is quasistatic, we dropped the time index for simplicity of notation. Because $r_{1,2} = r_{3,4} = 0$, the matrices $\mathbf{A}$ and $\mathbf{D}$ in (5.22) are diagonal with real entries. Specifically,

we have:

$$\mathbf{A} = \begin{bmatrix} r_{1,1} & r_{1,2} \\ 0 & r_{2,2} \end{bmatrix} = \begin{bmatrix} r_{1,1} & 0 \\ 0 & \frac{1}{\sqrt{3}} r_{1,1} \end{bmatrix} \text{ and } \mathbf{D} = \begin{bmatrix} r_{3,3} & r_{3,4} \\ 0 & r_{4,4} \end{bmatrix} = \begin{bmatrix} r_{3,3} & 0 \\ 0 & \frac{1}{\sqrt{3}} r_{3,3} \end{bmatrix}. \tag{D.3}$$

□

APPENDIX E

PROOF OF THE SEPARABILITY OF EMBEDDED ORTHOGONAL SPACE-TIME BLOCK CODE FOR $\mathcal{R} = \mathcal{R}_1$

We will determine the separability and the worst-case ML decoding complexity of the embedded orthogonal space-time block code by examining the properties of the $\mathbf{R}$ matrix in the QR decomposition of the effective channel matrix in (6.24). For the $\boldsymbol{\mathcal{G}}(M_1, \mathcal{R}_1)$ embedded code and for $\mathcal{R} = \mathcal{R}_1$, then it follows immediately that $\mathcal{R}_2 = \lceil \mathcal{R}/\mathcal{R}_1 \rceil = 1$ and $M_2 = M/M_1$. Without loss of generality, we can assume a single receive antenna such that $N = 1$. Let $\eta = 2\mathcal{R}_1 T_1$, then the effective channel matrix is given by

$$
\begin{aligned}
\check{\mathbf{H}} &= \texttt{blkdiag}\left(\bar{\mathbf{H}}_1\right) \bar{\mathbf{G}} \\
&= \begin{bmatrix} \boldsymbol{\mathcal{H}}_{1,1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{H}}_{1,2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\mathcal{H}}_{1,M_2} \end{bmatrix} (\mathbf{G} \otimes \mathbf{I}_\eta) \\
&= \underbrace{\begin{bmatrix} \boldsymbol{\mathcal{H}}_{1,1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{H}}_{1,2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\mathcal{H}}_{1,M_2} \end{bmatrix}}_{\texttt{blkdiag}\left(\bar{\mathbf{H}}_1\right)} \underbrace{\begin{bmatrix} g_{1,1}\mathbf{I}_\eta & g_{1,2}\mathbf{I}_\eta & \cdots & g_{1,M_2}\mathbf{I}_\eta \\ g_{2,1}\mathbf{I}_\eta & g_{2,2}\mathbf{I}_\eta & \cdots & g_{2,M_2}\mathbf{I}_\eta \\ \vdots & \vdots & \ddots & \vdots \\ g_{M_2,1}\mathbf{I}_\eta & g_{M_2,2}\mathbf{I}_\eta & \cdots & g_{M_2,M_2}\mathbf{I}_\eta \end{bmatrix}}_{\bar{\mathbf{G}}}.
\end{aligned} \tag{E.1}
$$

We next perform a QR decomposition on $\texttt{blkdiag}\left(\bar{\mathbf{H}}_1\right)$ such that $\texttt{blkdiag}\left(\bar{\mathbf{H}}_1\right) = \mathbf{Q}_H \mathbf{R}_H$. Because each subblock $\boldsymbol{\mathcal{H}}_{1,m}$, $m \in \{1, \cdots, M_2\}$ in the block diagonal matrix is

orthogonal, it follows that

$$\texttt{blkdiag}\left(\bar{\mathbf{H}}_1\right) = \mathbf{Q}_H \underbrace{\begin{bmatrix} \beta_1\mathbf{I}_\eta & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \beta_2\mathbf{I}_\eta & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \beta_{M_2}\mathbf{I}_\eta \end{bmatrix}}_{\mathbf{R}_H}, \tag{E.2}$$

where $\beta_m = \sqrt{\sum_{l=0}^{M_1-1} |h_{M_1 m-l,1}|^2}$, $m \in \{1, \cdots, M_2\}$.

The effective channel matrix in (E.1) can then be expressed as

$$\begin{aligned} \check{\mathbf{H}} &= \mathbf{Q}_H \begin{bmatrix} \beta_1\mathbf{I}_\eta & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \beta_2\mathbf{I}_\eta & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \beta_{M_2}\mathbf{I}_\eta \end{bmatrix} \begin{bmatrix} g_{1,1}\mathbf{I}_\eta & g_{1,2}\mathbf{I}_\eta & \cdots & g_{1,M_2}\mathbf{I}_\eta \\ g_{2,1}\mathbf{I}_\eta & g_{2,2}\mathbf{I}_\eta & \cdots & g_{2,M_2}\mathbf{I}_\eta \\ \vdots & \vdots & \ddots & \vdots \\ g_{M_2,1}\mathbf{I}_\eta & g_{M_2,2}\mathbf{I}_\eta & \cdots & g_{M_2,M_2}\mathbf{I}_\eta \end{bmatrix} \\ &= \mathbf{Q}_H \begin{bmatrix} \beta_1 g_{1,1}\mathbf{I}_\eta & \beta_1 g_{1,2}\mathbf{I}_\eta & \cdots & \beta_1 g_{1,M_2}\mathbf{I}_\eta \\ \beta_2 g_{2,1}\mathbf{I}_\eta & \beta_2 g_{2,2}\mathbf{I}_\eta & \cdots & \beta_2 g_{2,M_2}\mathbf{I}_\eta \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{M_2} g_{M_2,1}\mathbf{I}_\eta & \beta_{M_2} g_{M_2,2}\mathbf{I}_\eta & \cdots & \beta_{M_2} g_{M_2,M_2}\mathbf{I}_\eta \end{bmatrix} \\ &= \mathbf{Q}_H\mathbf{F}. \end{aligned} \tag{E.3}$$

Let $\mathbf{f}_m$, $m \in \{1, \cdots, \eta M_2\}$, designate the m-th column of the matrix $\mathbf{F}$. There are η orthogonal subspaces in the matrix $\mathbf{F}$. This follows from the simple form of the matrix $\mathbf{F}$, wherein each subblock is a scaled $\eta \times \eta$ identity matrix with all its columns orthogonal to each other. Specifically, the subspaces spanned by the group of columns $\boldsymbol{\mathcal{F}}_l = \left\{\mathbf{f}_{l+\eta(k-1)}\right\}$ for $l \in \{1, \cdots, \eta\}$ and $k \in \{1, \cdots, M_2\}$, are orthogonal to each other. We perform another QR decomposition; this time on the matrix $\mathbf{F}$ to obtain

$$\mathbf{F} = \mathbf{Q}_F\mathbf{R}, \tag{E.4}$$

where

$$\begin{aligned}
\mathbf{R} &= \begin{bmatrix} r_{1,1}\mathbf{I}_\eta & r_{1,2}\mathbf{I}_\eta & \cdots & r_{1,M_2}\mathbf{I}_\eta \\ & r_{2,2}\mathbf{I}_\eta & \cdots & r_{2,M_2}\mathbf{I}_\eta \\ & & \ddots & \vdots \\ & & & r_{M_2,M_2}\mathbf{I}_\eta \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,M_2} \\ & r_{2,2} & \cdots & r_{2,M_2} \\ & & \ddots & \vdots \\ & & & r_{M_2,M_2} \end{bmatrix}}_{\bar{\mathbf{R}}} \otimes \mathbf{I}_\eta \\
&= \bar{\mathbf{R}} \otimes \mathbf{I}_\eta. \qquad \text{(E.5)}
\end{aligned}$$

The form of the $\mathbf{R}$ matrix follows directly from the fact that there are η orthogonal subspaces. Substituting $\mathbf{F} = \mathbf{Q}_F\mathbf{R}$ into (E.3) yields the desired QR decomposition $\check{\mathbf{H}} = \mathbf{QR}$, where $\mathbf{Q} = \mathbf{Q}_H\mathbf{Q}_F$.

Using the real-valued system model in (2.7), the ML decoder minimizes the cost function

$$\begin{aligned}
P(\check{\mathbf{x}}) &= \|\check{\mathbf{y}} - \check{\mathbf{H}}\check{\mathbf{x}}\|^2 \\
&= \|\mathbf{z} - \mathbf{R}\check{\mathbf{x}}\|^2, \qquad \text{(E.6)}
\end{aligned}$$

where $\mathbf{z} = \mathbf{Q}^\top\check{\mathbf{x}}$ and $\mathbf{R}$ is given in (E.5). The cost function in (E.6) can be written as the sum of η cost functions as follows

$$\begin{aligned}
P(\check{\mathbf{x}}) &= \|\mathbf{z} - \mathbf{R}\check{\mathbf{x}}\|^2 \\
&= \sum_{l=1}^{\eta} \|\mathbf{z}_l - \bar{\mathbf{R}}\check{\mathbf{x}}_l\|^2, \qquad \text{(E.7)}
\end{aligned}$$

where

- $\mathbf{z}_l = \left\{z_{l+\eta(k-1)}\right\}$, $l \in \{1, \cdots, \eta\}$, and $k \in \{1, \cdots, M_2\}$
- $\check{\mathbf{x}}_l = \left\{\check{x}_{l+\eta(k-1)}\right\}$, $l \in \{1, \cdots, \eta\}$, and $k \in \{1, \cdots, M_2\}$
- $\bar{\mathbf{R}}$ is the $M_2 \times M_2$ upper triangular matrix in (E.5)

As can be seen from (E.7), the ML cost function can be written as the sum of η independent terms. Therefore, the decoding can be done over η independent groups, with each group containing M_2 real symbols. Hence, the rate $\mathcal{R} = \mathcal{R}_1$ embedded orthogonal space-time block code for M antennas is η-group decodable. Since $\eta = 2\mathcal{R}_1 T_1 \geq 2\min\{\mathcal{R}_1 T_1\} \geq 2 \times 2 = 4$, the embedded orthogonal space-time block code is separable.

Finally, since each group contains M_2 real symbols drawn from $\sqrt{q}$-ary alphabet, the worst-case ML decoding complexity is $\mathcal{O}\left(\sqrt{q}^{M_2-1}\right) = \mathcal{O}\left(q^{\frac{M_2-1}{2}}\right)$. This is because there are $\sqrt{q}^{M_2-1}$ ways to choose the first $M_2 - 1$ symbols, and for each choice, the last symbol can be decoded using a slicer, whose complexity is $\mathcal{O}(1)$. Since $M_2 = M/M_1$, we have that the ML decoding complexity is $\mathcal{O}\left(q^{\frac{M/M_1-1}{2}}\right)$. □

APPENDIX F

PROOF THAT THE EMBEDDED ORTHOGONAL SPACE-TIME BLOCK CODE IS NOT SEPARABLE FOR $\mathcal{R} > \mathcal{R}_1$

In order for the embedded orthogonal space-time block code to be separable, there needs to be at least one column in one group of columns in the effective channel matrix that is orthogonal to all the other columns in the other groups. For example, for a 2-group decodable space-time code, there needs to be two groups of columns, where the columns in the first group are orthogonal to all the columns in the second group.

It is sufficient to examine orthogonality between the columns in a simpler form of the effective channel matrix. Specifically, we can assume a single receive antenna such that the effective channel matrix is given by

$$\mathbf{H} = \left[\texttt{blkdiag}\left(\boldsymbol{\mathcal{J}}^0 \bar{\mathbf{H}}_1\right) \bar{\mathbf{G}} \quad \texttt{blkdiag}\left(\boldsymbol{\mathcal{J}}^1 \bar{\mathbf{H}}_1\right) \bar{\mathbf{G}} \quad \cdots \quad \texttt{blkdiag}\left(\boldsymbol{\mathcal{J}}^{\mathcal{R}_2-1} \bar{\mathbf{H}}_1\right) \bar{\mathbf{G}} \right], \quad \text{(F.1)}$$

where $\mathcal{R}_2 = \lceil \mathcal{R}/\mathcal{R}_1 \rceil \geq 2$. This simplification results from the fact that all subsequent rows in the effective channel matrix have identical form to (F.1), and they do not encode fading coefficients that appear in the first $2T_1M_2$ rows.

Since ϕ can take any value such that $|\phi| = 1$, we will simply assume that $\phi = 1$. The effective channel matrix in (F.1) can be written as

$$\mathbf{H} = \begin{bmatrix} g_{1,1}\boldsymbol{\mathcal{H}}_{1,1} & \cdots & g_{1,M_2}\boldsymbol{\mathcal{H}}_{1,1} & \cdots & g_{1,1}\boldsymbol{\mathcal{H}}_{1,M_2} & \cdots & g_{1,M_2}\boldsymbol{\mathcal{H}}_{1,M_2} \\ g_{2,1}\boldsymbol{\mathcal{H}}_{1,2} & \cdots & g_{2,M_2}\boldsymbol{\mathcal{H}}_{1,2} & \cdots & g_{2,1}\boldsymbol{\mathcal{H}}_{1,1} & \cdots & g_{2,M_2}\boldsymbol{\mathcal{H}}_{1,1} \\ \vdots & \cdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ g_{M_2,1}\boldsymbol{\mathcal{H}}_{1,M_2} & \cdots & g_{M_2,M_2}\boldsymbol{\mathcal{H}}_{1,M_2} & \cdots & g_{M_2,1}\boldsymbol{\mathcal{H}}_{1,M_2-1} & \cdots & g_{M_2,M_2}\boldsymbol{\mathcal{H}}_{1,M_2-1} \end{bmatrix}. \quad \text{(F.2)}$$

Separability for any space-time code should hold for all channel realizations. To show that the embedded orthogonal code is not separable, we only need to show that there is at least one channel realization for which the code is not separable. We will consider two cases,

when $M_2 = 2$ and when $M_2 > 2$. We do not consider the case $M_2 = 1$, since as discussed earlier, it implies that there is no embedding, and we simply obtain the orthogonal design with $\mathcal{R} = \mathcal{R}_1$. As proven in lemma 6.2, the embedded orthogonal design is separable for $\mathcal{R} = \mathcal{R}_1$. In this lemma, we are concerned with the case of $\mathcal{R} > \mathcal{R}_1$, which implies that $M_2 > 1$.

Case 1 ($M_2 = 2$): Since $M_1 \in \{2, 3, 4\}$ and $M_2 = 2$, we have that $M \in \{4, 6, 8\}$. Given the finite number of configurations, one can easily verify numerically that the embedded orthogonal space-time block code is not separable from the form of the $\mathbf{R}$ matrix in the QR decomposition of the effective channel matrix.

Case 2 ($M_2 > 2$): Let us assume a channel realization wherein $\boldsymbol{\mathcal{H}}_{1,3}$ through $\boldsymbol{\mathcal{H}}_{1,M_2}$ are all zero. With this assumption, let us consider the orthogonality among M_2 groups of columns, where the ℓ-th group, $\ell \in \{0, \cdots, \mathcal{R}_2 - 1\}$, contains the columns of $\texttt{blkdiag}\left(\boldsymbol{\mathcal{J}}^{\ell}\bar{\mathbf{H}}_1\right)\bar{\mathbf{G}}$. For a column in the ℓ_1-th group to be orthogonal to a column in the ℓ_2-th group, where $\ell_1 \neq \ell_2$, the product $\mathbf{h}_{k_1,1}^{\top}\mathbf{h}_{k_2,2}$ must be zero, where $\mathbf{h}_{k_1,1}$ is the k_1-th column in $\boldsymbol{\mathcal{H}}_{1,1}$ and $\mathbf{h}_{k_2,2}$ is the k_2-th column in $\boldsymbol{\mathcal{H}}_{1,2}$. This follows directly from the effective channel matrix in (F.2) and from the assumption that $\boldsymbol{\mathcal{H}}_{1,3}$ through $\boldsymbol{\mathcal{H}}_{1,M_2}$ are zero. For example, for $M_2 = 3$ and $R_2 = 2$, we have

$$\mathbf{H} = \begin{bmatrix} g_{1,1}\boldsymbol{\mathcal{H}}_{1,1} & g_{1,2}\boldsymbol{\mathcal{H}}_{1,1} & g_{1,3}\boldsymbol{\mathcal{H}}_{1,1} & g_{1,1}\boldsymbol{\mathcal{H}}_{1,2} & g_{1,2}\boldsymbol{\mathcal{H}}_{1,2} & g_{1,3}\boldsymbol{\mathcal{H}}_{1,2} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ g_{2,1}\boldsymbol{\mathcal{H}}_{1,2} & g_{2,2}\boldsymbol{\mathcal{H}}_{1,2} & g_{2,3}\boldsymbol{\mathcal{H}}_{1,2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & g_{2,1}\boldsymbol{\mathcal{H}}_{1,1} & g_{2,2}\boldsymbol{\mathcal{H}}_{1,1} & g_{2,3}\boldsymbol{\mathcal{H}}_{1,1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & g_{3,1}\boldsymbol{\mathcal{H}}_{1,1} & g_{3,2}\boldsymbol{\mathcal{H}}_{1,1} & g_{3,3}\boldsymbol{\mathcal{H}}_{1,1} & g_{3,1}\boldsymbol{\mathcal{H}}_{1,2} & g_{3,2}\boldsymbol{\mathcal{H}}_{1,2} & g_{3,3}\boldsymbol{\mathcal{H}}_{1,2} \end{bmatrix}. \tag{F.3}$$

The product $\mathbf{h}_{k_1,1}^{\top}\mathbf{h}_{k_2,2}$ is in general nonzero. Therefore, there is no column in the ℓ_1-th group that is orthogonal to any column in the ℓ_2-th group, where $\ell_1 \neq \ell_2$. Therefore, we cannot construct a group of columns that contains columns from the different M_2 groups such that its columns are orthogonal to the other groups of columns. Therefore, the embedded orthogonal space-time block code is not separable for $M_2 > 2$.

The embedded orthogonal space-time block code is not separable for $M_2 > 1$. Therefore, the embedded orthogonal code is not separable for $\mathcal{R} > \mathcal{R}_2$, since this relationship implies that $M_2 > 1$. □

REFERENCES

[1] M. Mouly and M.-B. Pautet, *The GSM System for Mobile Communications*, Telecom Publishing, June 1992.

[2] G. J. Foschini and M. J. Gans, "On Limits of Wireless Communication in a Fading Environment When Using Multiple Antennas," *Wireless Personal Commun.*, pp. 36-54, March 1998.

[3] L. Zheng and D. Tse, "Diversity and Multiplexing: A Fundamental Tradeoff in Multiple Antenna Channels," *IEEE Trans. on Inf. Theory*, vol. 49, no 4, pp. 1073-1096, May 2003.

[4] M. O. Damen, H. E. Gamal and N. C. Beaulieu, "Linear Threaded Algebraic Space-Time Constellations," *IEEE Trans. Inf. Theory*, vol. 49, pp. 2372-2388, Oct. 2003.

[5] H. E. Gamal and M. O. Damen, "Universal Space-Time Coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1097-1119, May 2003.

[6] J. H. Sung, "Transmitter Strategies for Closed-Loop MIMO-OFDM," Ph.D. Thesis, Georgia Institute of Technology, 2004.

[7] J.-C. Belfiore, G. Rekaya, and E. Viterbo, "The Golden Code: A 2×2 Full Rate Space-Time Code with Non Vanishing Determinants," *IEEE Trans. on Inf. Theory*, vol. 51, no. 4, April 2005.

[8] P. Dayal and M. K. Varanasi, "An Optimal Two Transmit Antenna Space-Time Code And Its Stacked Extensions," *IEEE Trans. on Inf. Theory*, vol. 51, pp. 4348–4355, Dec. 2005.

[9] H. Yao and G. W. Wornell, "Achieving the Full MIMO Diversity-Multiplexing Frontier with Rotation-Based Space-Time Codes," in *Proceedings Allerton Conf. Commun., Cont., and Computing*, (Illinois), Oct. 2003.

[10] IEEE 802.16e-2005: IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment 2: Physical Layer and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, Feb. 2006.

[11] B. Cerato, G. Masera, and E. Viterbo, "A VLSI Decoder for the Golden Code," 13th *IEEE International Conference on Electronics, Circuits and Systems*, ICECS '06, pp. 549 - 552, Dec. 10-13, 2006.

[12] L. Zhang, B. Li, T. Yuan, X. Zhang, and D. Yang, "Golden Code with Low Complexity Sphere Decoder," *18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)*, pp. 1-5, Athens, Sept. 3-7, 2007.

[13] M. Sarkiss, J.-C. Belfiore, and Y.-W. Yi, "Performance Comparison of Different Golden Code Detectors," *18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)*, pp. 1-5, Athens, Sept. 3-7, 2007.

[14] S. D. Howard, S. Sirianunpiboon, and A. R. Calderbank, "Fast Decoding of the Golden Code by Diophantine Approximation," *IEEE Information Theory Workshop*, Lake Tahoe, California, pp. 590-594, Sept. 2-6, 2007.

[15] S. Sirianunpiboon, A. R. Calderbank and S. D. Howard, "Fast Essentially Maximum Likelihood Decoding of the Golden Code," submitted to *IEEE Trans. on Inf. Theory*, 2008.

[16] O. Tirkkonen and R. Kashaev, "Combined Information and Performance Optimization of Linear MIMO Modulations," in *Proc IEEE Int. Symp. Inf. Theory (ISIT 2002)*, Lausanne, Switzerland, p. 76, June 2002.

[17] J. Paredes, A. B. Gershman, and M. G. Alkhanari, "A New Full-Rate Full-Diversity Space-Time Block Code With Nonvanishing Determinant and Simplified Maximum-Likelihood Decoding," *IEEE Trans. on Signal Processing*, vol. 56, pp. 2461-2469, June 2008.

[18] S. Sezginer and H. Sari, "Full-Rate Full-Diversity 2×2 Space-Time Codes for Reduced Decoding Complexity," *IEEE Commun. Letters*, vol. 11, no. 12, pp. 1-3, Dec. 2007.

[19] F. Oggier, G. Rekaya, J.-C. Belfiore, and E. Viterbo, "Perfect Space-Time Block Codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 3885-, Sep. 2006.

[20] P. Elia, B. A. Sethuraman and P. Kumar, "Perfect Space-Time Codes with Minimum And Non-Minimum Delay For Any Number of Transmit Antennas, *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 3853-, Nov. 2007.

[21] S. M. Alamouti, "A Simple Transmit Diversity Technique For Wireless Communications," *IEEE J. Sel. Areas Commun.*, vol. 16, pp. 1451-1458, Oct. 1998.

[22] N. Sharma and C. Papadias, "Full Rate Full Diversity Linear Quasi-Orthogonal Space-Time Codes For Any Transmit Antennas," *EURASIP Journal Applied Signal Processing*, no. 9, pp. 1246-, Aug. 2004.

[23] D. Dao, C. Yuen, C. Tellambura, Y. Guan, T. T. Tjhung, "Four-Group Decodable Space-Time Block Codes," *IEEE Trans. Signal Processing*, vol. 56, pp. 424-430, Jan. 2008.

[24] E. Biglieri, Y. Hong, and E. Viterbo, "On Fast Decodable Space-Time Block Codes," *IEEE Trans. on Inf. Theory*, vol. 55, no. 2, pp. 524-530, Feb. 2009.

[25] V. Tarokh, H. Jafarkhani and A. Calderbank, "Space-Time Block Codes From Orthogonal Designs," *IEEE Trans. on Inf. Theory*, vol. 45, no. 5, pp. 1456-1467, July 1999.

[26] X.-B. Liang, "Orthogonal Designs with Maximal Rates," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 24682503, Oct. 2003.

[27] M. O. Damen, K. Abed-Meraim, and J.-C. Belfiore, "Diagonal Algebraic Space-Time Block Codes," *IEEE Trans. on Inf. Theory*, vol. 48, no. 3, pp. 628-636, Mar. 2002.

[28] M. O. Damen and N. C. Beaulieu, "On Diagonal Algebraic Space-Time Block Codes," *IEEE Trans. Commun.*, vol. 51, no. 6, pp. 911-919, June 2003.

[29] M. O. Sinnokrot and J. R. Barry, "The Golden Code is Fast Decodable," *IEEE Global Telecommunications Conference*, (Globecom 2008), New Orleans, pp. 1-5, Nov. 30-Dec. 4, 2008.

[30] M. O. Sinnokrot and J. R. Barry, "Fast Maximum-Likelihood Decoding of the Golden Code," accepted, to appear, *IEEE Trans. on Wireless Commun.*, 2009.

[31] M. O. Sinnokrot and J. R. Barry, "Modified Golden Codes for Fast Decoding on Time-Varying Channels," *The 11th International Symposium on Wireless Personal Multimedia Communications* (WPMC 2008), Lapland, Finland, Sep. 2008.

[32] M. O. Sinnokrot, J. R. Barry and V. K. Madisetti, "The Asymmetric Golden Code for Fast Decoding on Time-Varying Channels," to appear, *Wireless Personal Communications*, 2009.

[33] M. O. Sinnokrot, J. R. Barry and V. K. Madisetti, "Embedded Alamouti Space-Time Codes for High Rate and Low Decoding Complexity," *Asilomar Conference on Signals, Systems, and Computers*, (Asilomar 2008), Pacific Grove, California, Oct. 26-29, 2008.

[34] M. O. Sinnokrot, J. R. Barry, and V. K. Madisetti, "Embedded Orthogonal Space-Time Codes for High Rate and Low Decoding Complexity," to appear, *IEEE Global Communications Conference* (Globecom 2009), Honolulu, Hawaii, Nov. 30-Dec. 4, 2009.

[35] A. Wittneben, "A New Bandwidth Efficient Transmit Antenna Modulation Diversity Scheme for Linear Digital Modulation," in *IEEE International Conference on Communications*, vol. 3, pp. 1630-1634, Geneva, Switzerland, May 1993.

[36] N. Seshadri and J. H. Winters, "Two Schemes for Improving the Performance of Frequency-Division Duplex (FDD) Transmission Systems Using Transmitter Antenna

Diversity," *International Journal of Wireless Information Networks*, vol. 1, pp. 49-60, Jan. 1994.

[37] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction," *IEEE Trans. Inf. Theory*, vol. 44, pp. 744-765, Mar. 1998.

[38] H. Jafarkhani, "A Quasi-orthogonal Space-Time Block Code," *IEEE Trans. Commun.*, vol. 49, no. 1, pp. 14, Jan. 2001.

[39] O. Tirkkonen, A. Boariu, and A. Hottinen, "Minimal Nonorthogonality Rate 1 Space-Time Block Code For 3+ Tx Antennas," in *Proc. IEEE 6th Int. Symp. Spread-Spectrum Techniques and Applications (ISSSTA)*, pp. 429-432, Sept. 2000.

[40] C. B. Papadias and G. J. Foschini, "Capacity-Approaching Space-Time Codes For Systems Employing Four Transmitter Antennas," *IEEE Trans. on Inf. Theory*, vol. 49, No. 3, pp. 726-732, Mar. 2003.

[41] O. Tirkkonen, "Optimizing Space-Time Block Codes By Constellation Rotations," in *Proc. Finnish Wireless Commun. Workshop (FWCW)*, Finland, pp. 59-60, Oct. 2001.

[42] N. Sharma and C. Papadias, "Improved Quasi-orthogonal Codes Through Constellation Rotation," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 332-335, Mar. 2003.

[43] W. Su and X.-G. Xia, "Signal Constellations For Quasi-orthogonal Space-Time Block Codes With Full Diversity," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2331-2347, Oct. 2004.

[44] D. Tse and P. Viswanath, *Fundamentals of wireless communication*, Cambridge University Press, Cambridge, UK, 2005.

[45] B. Hassibi and B. M. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Inf. Theory*, vol. 48, pp. 1804-1824, July 2002.

[46] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications," *ACM SIGSAM Bull.*, vol. 15, pp. 37-44, Feb. 1981.

[47] C. P. Schnorr, and M. Euchner, "Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems," *Math. Programming*, vol. 66, pp. 181-191, 1994.

[48] 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, July 1999.

[49] M. O. Damen, A. Chkeif, and J. C. Belfiore, "Lattice Code Decoder for Space-Time Codes," *IEEE Commun. Letters*, pp. 161-163, May 2000.

[50] M. K. Simon and J. G. Smith, "Hexagonal Multiple Phase-and-Amplitude-Shift-Keyed Signal Sets," *IEEE Trans. on Commun.*, vol. 21, no. 10, pp. 1108-1115, Oct. 1973.

[51] G. D. Forney, R. G. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Qureshi, "Efficient Modulation for Band-Limited Channels," *IEEE J. Sel. Areas Commun.*, vol. 2, pp. 632-647, Sep. 1984.

[52] J. R. Barry, E. A. Lee, D. G. Messerschmitt, *Digital Communication*, Kluwer Academic Publishers, 3rd ed., 2003.

[53] P. Elia, K. R. Kumar, S. A. Pawar, P. V. Kumar and H.-F. Lu, "Explicit, Minimum-Delay Space-Time Codes Achieving The Diversity-Multiplexing Gain Tradeoff," *IEEE Trans. on Inf. Theory*, vol. 52, pp. 3869-3884, Sep. 2006.

[54] G. Ganesan and P. Stoica, "Space-Time Block Codes: A Maximum SNR Approach," *IEEE Trans. on Inf. Theory*, vol. 47, pp. 16501656, May 2001.

[55] Z. Li-gang, M. Jian-song, L. Xin and D. Wei, "Complete Design for Complex Orthogonal Designs with Maximal Rates and Minimal Decoding Delays," *6th International Conference on ITS Telecommunications Proceedings*, pp. 390-393, June 2006.

[56] J. Boutros and E. Viterbo, "Signal Space Diversity: A Power and Bandwidth Efficient Diversity Technique for the Rayleigh Fading Channel," *IEEE Trans. Inf. Theory*, vol. 44, pp. 1453-1467, July 1998.

[57] J.-C. Belfiore, X. Giraud, and J. Rodriguez, "Linear Labeling for Joint Source Channel Coding," in *Proc. Int. Symp. Information Theory (ISIT)*, Sorrento, Italy, June 2000.

[58] C. Yuen, Y. L. Guan, and T. T. Tjhung, "Quasi-Orthogonal STBC With Minimum Decoding Complexity," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2089-2094, Sep. 2005.

[59] M. Z. A. Khan and B. S. Rajan, "Single-Symbol Maximum Likelihood Decodable Linear STBCs," *IEEE Trans. on Inf. Theory*, vol. 52, no. 5, pp. 2062-2091, May 2006.

[60] P. Marsch, W. Rave and G. Fettweis, "Quasi-Orthogonal STBC Using Stretched Constellations For Low Detection Complexity," in *Proc. Wireless Communications and Networking Conf. (WCNC)*, pp. 757-761, Mar. 2007.

[61] M. O. Sinnokrot and J. R. Barry, "A Single-Symbol Decodable Space-Time Block Code with Full Rate and Low Peak-to-Average-Power Ratio," accepted, to appear, *IEEE Trans. on Wireless Commun.*, April 2008.

[62] F. Oggier and E. Viterbo, *Full Diversity Rotations*. [Online]. Available: www1.tlc.polito.it/∼viterbo/rotations/rotations.html.

[63] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, 1997.

[64] W. Gander, "Algorithms for the QR Decomposition," *Research report No. 80-02, Eidgenoessische Technische Hochschule*, CH-8092 Zuerich, 1980.

[65] A. Wiesel, X. Mestre, A. Pages and J. R. Fonollosa, "Efficient Implementation of Sphere Demodulation," *4th IEEE Workshop on Signal Processing Advances in Wireless Communications, (SPAWC 03)*, Rome, Italy, pp. 36-40, June 15-18, 2003.

[66] A. Ismail, H. Sari, J. Fiorina and M. O. Damen, "A Rate 3/2 Full-Diversity 4×4 Space-Time Code with Fast Maximum-Likelihood Decoding," *20th Annual IEEE*

International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'09), Tokyo, Japan, Sept. 13-16, 2009.

[67] C. Yuen, Y. L. Guan, T. T. Tjhung, "On the Search for High-Rate Quasi-Orthogonal Space-Time Block Code," *International Journal of Wireless Information Network*, vol. 13, no. 4, pp. 329-340, Oct. 2006.

[68] J. Chung, S. H. Nam and C.-S. Hwang, "High Rate Space-Time Block Codes," *IEICE Trans. on Commun.*, vol. E89-B, no. 4, pp. 1420-1422, April 2006.

[69] H. Yao, and G. W. Wornell, "Lattice-Reduction-Aided Detectors for MIMO Communication Systems," *IEEE Global Telecommun. Conf.*, vol. 1, pp. 424-428, Nov. 2002.

[70] Z. Guo and P. Nilsson, "Algorithm and Implementation of the K-Best Sphere Decoding for MIMO Detection," *IEEE Journal on Selected Areas on Communications*, vol. 24, pp. 491-503, Mar. 2006.

[71] L. G. Barbero and J. S. Thompson, "A Fixed-Complexity MIMO Detector Based on the Complex Sphere Decoder," in *IEEE International Workshop on Signal Processing Advances in Wireless Communications* (SPAWC '06), Cannes, France, July 2006.

[72] D. W. Waters, "Signal Detection Strategies and Algorithms for Multiple-Input Multiple-Output Channels," Ph.D. Thesis, Georgia Institute of Technology, 2005.

[73] W. Zhang and X.-G. Xia, "A Design of Space-Time Block Codes Achieving Full Diversity with Partial Interference Cancellation Group Decoding," Submitted to *IEEE Trans. Inf. Theory*, available online at arXiv , arXiv:0904.1812v2 [cs.IT].

VITA

Mohanned Sinnokrot was born in Mafraq, Jordan in 1978. He grew up in Amman, Jordan, and graduated from high school in 1995 from Doha, Qatar. He achieved the highest score on the nationwide college entrance examination in Qatar, and he was awarded a scholarship from the government of Qatar to attend Georgia Institute of Technology. In May 2001, he received a bachelor of science degree in Biochemistry and a bachelor of science degree in Electrical and Computer Engineering from Georgia Institute of Technology; both degrees with highest honor.

He joined Telematix in 2001, as a digital design engineer working on algorithm design, implementation and verification for wireless local area network communication chips. He later left Telematix in 2004 to pursue his graduate studies in Electrical and Computer Engineering at the Georgia Institute of Technology.

In the Summer of 2005, he was awarded a master of science degree in Electrical and Computer Engineering from the Georgia Institute of Technology with a specialization in Telecommunications. From 2005-2009, he worked under the supervision of Dr. Vijay Madisetti and Dr. John Barry towards his doctorate in the area of low complexity transceiver design for multiple antenna systems. In December 2009, Mr. Sinnokrot received his Ph.D in Electrical and Computer Engineering from the Georgia Institute of Technology.