University of California

Los Angeles

An Analytical Packet Error Rate Prediction for Punctured Convolutional Codes and an Application to CRC Code Design

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical Engineering

by

Chung-Yu Lou

2015

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Abstract of the Dissertation

An Analytical Packet Error Rate Prediction for Punctured Convolutional Codes and an Application to CRC Code Design

by

Chung-Yu Lou

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2015

Professor Babak Daneshrad, Chair

The performance of a packet-based communication system is determined by its packet error rate (PER) rather than its bit error rate (BER). Although the PER and the BER are correlated, it is not straightforward to calculate the PER using its BER due to the bit errors' correlation brought about by the channel coding. Therefore, the existing PER predictions are mostly heuristic methods.

The heuristic PER prediction methods, such as exponential effective SNR mapping (EESM) and mean mutual information per bit (MMIB), follow two steps: 1) compute an averaged single metric from the channel coefficients, and 2) use a simulated curve to map from the single metric to the PER. These methods require off-line simulations, curve-fitting, and parameter calibrations for every combination of modulation, code rate, and packet length. These requirements lower their applicability as modern systems support numerous modes (combinations of the above) thus needing an impractical number of off-line simulations. In this dissertation, we develop an analytical PER prediction method that requires no off-line simulations but delivers high accuracy for a wide range of transmission schemes and environments.

First, we assume that the system uses a uniform random interleaver and treat the bits' log-likelihood ratios (LLRs) as i.i.d. random variables. The LLR distribution is modeled as a mixture of folded normal distributions and approximated by a mixture of normal distributions. The PER is then calculated using a Gaussian Q-function approximation and the transfer function of the punctured convolutional code. Second, we focus on a realistic multiple-input multiple-output (MIMO) orthogonal frequency-division multiplexing (OFDM) system that uses a repeated-pattern interleaver. Therefore, the LLRs are dependent and their correlation is governed by their locations in the packet. In general, there are three types of correlations: a) correlations across subcarriers, b) correlations across spatial streams, and c) correlations within a subcarrier. All the three types of correlations are essential and their effect is characterized in our PER prediction. The simulation result shows that the analytical method achieves higher prediction accuracy than all the existing heuristic methods.

The analytical PER prediction can be applied to the design of cyclic redundancy check (CRC) code, which is used for error detection. We consider a system employing a CRC code concatenated with a convolutional code. The undetected error probability of this system can be calculated by two methods: the exclusion method and the construction method. The exclusion method enumerates the error patterns of the convolutional code and tests if each of them is detectable. The construction method reduces complexity significantly by exploiting the equivalence of the undetected error probability to the frame error rate of an equivalent catastrophic convolutional code. We further propose a design of the CRC code for a specific convolutional code and codeword length such that the undetected error probability is minimized. This probability can be minimized when the CRC code detects the most probable residual error patterns at the output of the Viterbi decoder. In our example, the designed CRC codes have significant reduction in undetected error probability compared to the existing CRC codes.

The dissertation of Chung-Yu Lou is approved.

Mario Gerla

Richard D. Wesel

Babak Daneshrad, Committee Chair

University of California, Los Angeles

2015

To my wife, Pei-Yun

Table of Contents

List of Figures

List of Tables

Acknowledgments

I want to thank my advisor, Professor Babak Daneshrad, for his guidance, patience, and encouragement throughout these years. He broadens my horizons and allows me to grow as an independent researcher. I would also like to extend my appreciation to the committee members, Professor Richard D. Wesel, Professor Kung Yao, and Professor Mario Gerla, for their precious time and insightful comments. Special thanks go to Professor Richard D. Wesel for his guidance during the research of cyclic redundancy check (CRC) codes in Chapter 4. In addition, his course Channel Coding Theory at UCLA has given me fundamental knowledge that is essential to this dissertation.

I am grateful to my excellent colleagues, Eren Eraslan and Chao-Yi Wang, in Wireless Integrated Systems Research Group at UCLA. Our collaborative project has led me to the entrance of this dissertation. All of our discussions and share of thoughts are valuable to me. I would also like to thank Adam R. Williamson and Kasra Vakilinia from Communications Systems Laboratory at UCLA for the useful discussions and technical help in the CRC code study in Chapter 4 and other projects.

Finally, I want to thank my family for their unconditional love and support in numerous ways. I would also like to express my deepest gratitude to my wife, Pei-Yun Lu, for her consideration, tolerance, and warm encouragement. This dissertation would not have been completed without her.

Vita

2006 B.S. (Electrical Engineering, minor in Mathematics), National Taiwan University, Taipei, Taiwan.

2009 M.S. (Electrical Engineering), UCLA, Los Angeles, California.

2010–2014 Graduate Student Researcher, Electrical Engineering Department, UCLA.

2010–2014 Teaching Assistant, Electrical Engineering Department, UCLA. Taught sections of Data Communications and Telecommunication Networks, Applied Numerical Computing, Digital Signal Processing and Signals and Systems.

2010 Engineer Intern, Silvus Technologies, Los Angeles, California.

2011 Graduate Intern Technical, Intel Corporation, Hillsboro, Oregon.

Publications

C.-Y. Lou and B. Daneshrad, "Analytical packet error rate predictions for punctured convolutional codes with bit-interleaved coded modulation," to be submitted to *IEEE Transactions on Vehicular Technology.*

C.-Y. Lou, B. Daneshrad, and R. D. Wesel, "Convolutional-code specific CRC code design to minimize the probability of undetected error," submitted to *IEEE Transactions on Communications.*

C.-Y. Lou, B. Daneshrad, and R. D. Wesel, "Optimizing pilot length for a go/no-go decision in two-state block fading channels with feedback," accepted by 2015 IEEE International Conference on Communications (ICC).

E. Eraslan, B. Daneshrad, and *C.-Y. Lou*, "Performance indicator for MIMO MMSE receivers in the presence of channel estimation error," *IEEE Wireless Communications Letter*, vol. 2, no. 2, pp. 211-214, Apr. 2013.

C.-Y. Lou and B. Daneshrad, "PER prediction for convolutionally coded MIMO OFDM systems—an analytical approach," *Proceedings of 2012 IEEE Military Communications Conference (MILCOM)*, Oct. 2012, pp. 1-6.

E. Eraslan, B. Daneshrad, *C.-Y. Lou*, and C. Wang, "Energy-aware fast link adaptation for a MIMO enabled cognitive system," *Green IT: Technologies and Applications*, J. H. Kim and M. J. Lee, Springer, 2011, ch. 3, pp. 37-55.

CHAPTER 1

Introduction

1.1 Background and Literature Review

1.1.1 Analytical Packet Error Rate Prediction

Modern radios are capable of transmitting and receiving in various schemes, which include the combinations of modulation, channel coding, transmit power. For a multiple-input multiple-output (MIMO) radio, its schemes even include the selections of antennas and MIMO techniques (beamforming, spatial multiplexing, or space-time code). Finding the optimal transmission scheme given an environment under the constraint of certain quality of service (QoS) is called the link adaptation problem [DCH10, KD10]. One of the solutions to the link adaptation problem is to find the link performance of each candidate transmission scheme and then pick the best one. Therefore, accurate link performance prediction plays an important role in the link adaptation problem.

Another demand of the link performance prediction comes from network-level simulations. Typically, a network-level simulation involves interactions between a large number of radios, e.g., mobile terminals, relays, and base stations. It is time consuming and impractical to include sample-accurate simulation of each transmission pair. This situation necessitates a simple and efficient way to characterize the link performance of the physical layer (PHY) [BM09].

The link performance of a packet-based communication system is exactly its packet error rate (PER) (frame error rate — FER, or block error rate — BLER).

In this dissertation, we assume that a packet contains only one codeword, and hence the PER and the codeword error rate are the same. In the case when a packet consists of multiple codewords, one can assume the codeword errors to be independent and thus calculate the PER based on the individual codeword error rates.

Convolutional codes and punctured convolutional codes are used extensively in wireless communication systems. Unlike turbo codes or low-density parity-check (LDPC) codes, (punctured) convolutional codes are not block codes. In other words, (punctured) convolutional codes can work with any length of input sequences and do not have a block structure. Thus, the PER of (punctured) convolutional codes gets scarcely comparable attention to their bit error rate (BER).

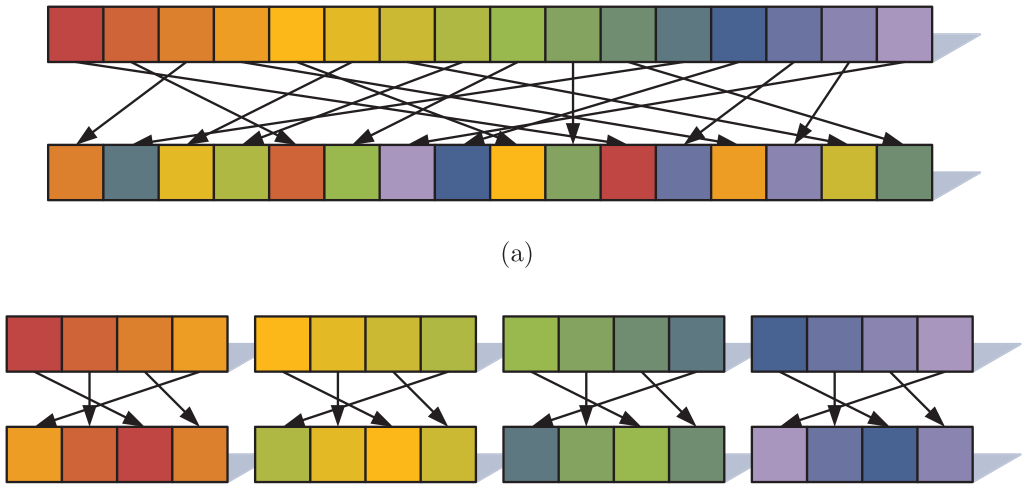
The BER of punctured convolutional codes are well-studied[Vit71, CCG79]. Although the PER and the BER are highly correlated, it is not straightforward to calculate the PER using its BER due to the bit errors' correlation brought about by the channel coding. Some papers[TWS08, EWD14] use the BER to calculate the PER by assuming that the bit errors are independent, but the results are not accurate.

The PER analysis of convolutional codes can be found in the literature. Hard decision was often assumed[PT87] but performs worse than soft decision. The code termination technique[KS05] treated the decoder trellis as a renewal process but required the bits in a codeword to have the same reliability. In practice, this requirement is rarely satisfied due to signal-to-noise ratio (SNR) variation in a codeword and high order quadrature amplitude modulation (QAM). The limiting before averaging (LBA) technique[ML99] provided an easy solution but only allowed a limited number of SNR levels in a codeword and binary phase-shift keying (BPSK). The PER of a convolutional code was also considered in the performance analysis of bit-interleaved coded modulation (BICM). The BICM expurgated bound[CTB98], refined from the union bound, only considered the nearest neighbor codeword and assumed that each bit of a codeword belongs to different modulation symbols. A more general approach in [YZS06] provided an accurate result but required numerical integration of Gaussian Q-function for each possible way of error bits to appear in a packet.

Most of the studies on BICM assume a uniform random interleaver or an infinite-size interleaver, also known as BICM with a single interleaver (BICM-S). The uniform random interleaver randomly creates a permutation for every transmission, and all permutations are equally probable. If the size of such interleaver is small ($<$ 1000 bits), the probability of two bits mapped to the same modulation symbol could be non-negligible. This creates correlation among bits' log-likelihood ratios (LLRs). The authors of [MF09] have demonstrated and analyzed this effect in a QPSK system operating in fading channels. Ideally, if the size of the uniform random interleaver is big enough, this interleaver approaches an infinite-size interleaver, which maps the neighbor encoded bits to distinct modulation symbols and independent channel fades. These properties simplify the analysis of the performance of such systems.

Recently, a few papers consider BICM with multiple interleavers (BICM-M). In BICM-M, each of the encoder output has its own infinite-size interleaver, and the bits from one encoder output must be assigned to a specific significance in the modulation. For example, a rate-1/2 convolutional encoder has two output bits, and they must be assigned to the most significant bit and the least significant bit of 16-QAM symbols, respectively. The BER analysis and design of BICM-M systems in non-fading channels is presented in [AAS10]. Another interleaver setting is BICM with a trivial interleaver (BICM-T). Similar to BICM-M, the bits from one encoder output in BICM-T must be assigned to a specific significance in the modulation. Unlike BICM-M, no interleaver is used in BICM-T. The BER analysis of BICM-T systems in non-fading channels is presented in [ASA11, MHA14].

The uniform random interleavers or infinite-size interleavers in BICM-S or



(a)

(b)

Figure 1.1: The illustrations of a uniform random interleaver (a) and a repeated–pattern interleaver (b).

BICM-M cannot be realized in a practical system. Consider a practical MIMO orthogonal frequency-division multiplexing (OFDM) system, such as Wi-Fi [Sta09]. It uses a fixed-pattern interleaver, which means that the permutation is the same for all transmissions. Moreover, the interleaver has a finite block size equal to the number of bits in an OFDM symbol, and thus all OFDM symbols have the same permutation pattern. We call this type of interleavers as repeated-pattern interleavers. Fig. 1.1 illustrates the difference between a uniform random interleaver and an repeated-pattern interleaver, where the repeated-pattern interleaver has a size of four bits.

Although the purpose of the interleaver is to avoid bursty errors of the encoded bits, a smaller size of bursty errors can still happen to the systems employing a repeated-pattern interleaver with certain transmission schemes in certain environments. The analysis of BICM-T [ASA11, MHA14] has similar considerations but is limited to single-input single-output (SISO) systems in non-fading channels. The authors of [DCH10] mentioned, “jointly modeling the effects of MIMO processing, OFDM modulation, convolutioal coding, and bit interleaving is challenging,” which is what we are presenting in this dissertation.

In addition to analytical approaches, heuristic approaches have been studied. In order to cope with SNR variation in a codeword, the concept of effective SNR was proposed in [NR98]. The effective SNR is obtained by properly averaging all the SNRs so that the system can be essentially viewed as operating in an additive white Gaussian noise (AWGN) channel at the effective SNR level and having the same error performance. Recently, several averaged metrics have been proposed and known as link quality metric (LQM), channel quality indicator (CQI), or PHY abstraction. Exponential effective SNR mapping (EESM)[Eri03] was based on Chernoff bound of Gaussian Q-function. Cumulant generating function based effective SNR mapping (κESM)[SGL09] was founded on the cumulant-generating function of the bits' LLRs. Mutual information effective SNR mapping (MIESM)[TS03] was calculated through averaged mutual information. Received bit information rate (RBIR)[WTA06] and mean mutual information per bit (MMIB)[SZS07] were based on averaged mutual information at the symbol-level and at the bit-level, respectively. Mean mutual information per coded bit mapping (MMIBM)[JKW10] improved MMIB by utilizing the variance of the bit-level mutual information as a correction term. However, since the relationship between PER and LQM is not straightforward, all LQMs rely on pre-simulated AWGN curves to map from the LQM to the PER.

In general, the heuristic methods can be described as the following. The post-processing SNR (PPSNR)[HSP01] values γ_i of all channel fades or subcarriers are the input, where i is the index of the channel coefficients. The PPSNRs are averaged through certain function $\mathrm{g}(\cdot)$ and yield a scalar metric

$$\mathrm{LQM} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{g}\left(\frac{\gamma_i}{\beta_{M,R,L}}\right), \tag{1.1}$$

where N is the total number of channel coefficients, M specifies the modulation scheme, R specifies the code rate, L represents the information length of the

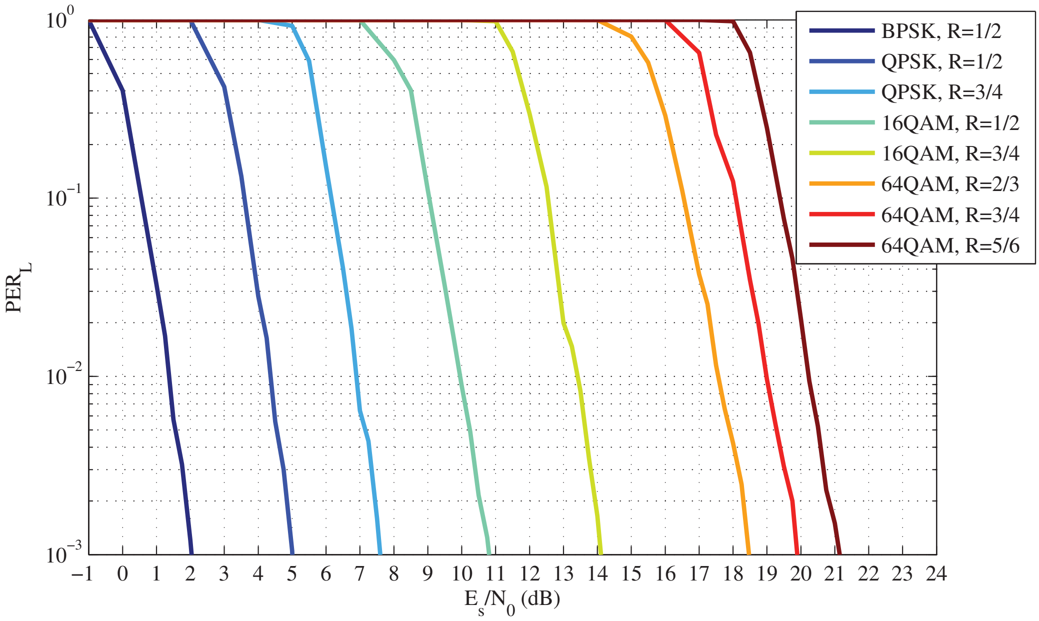


Figure 1.2: Simulated PER curves of various transmission schemes in an AWGN channel. The packet length is $L = 1024$ bytes.

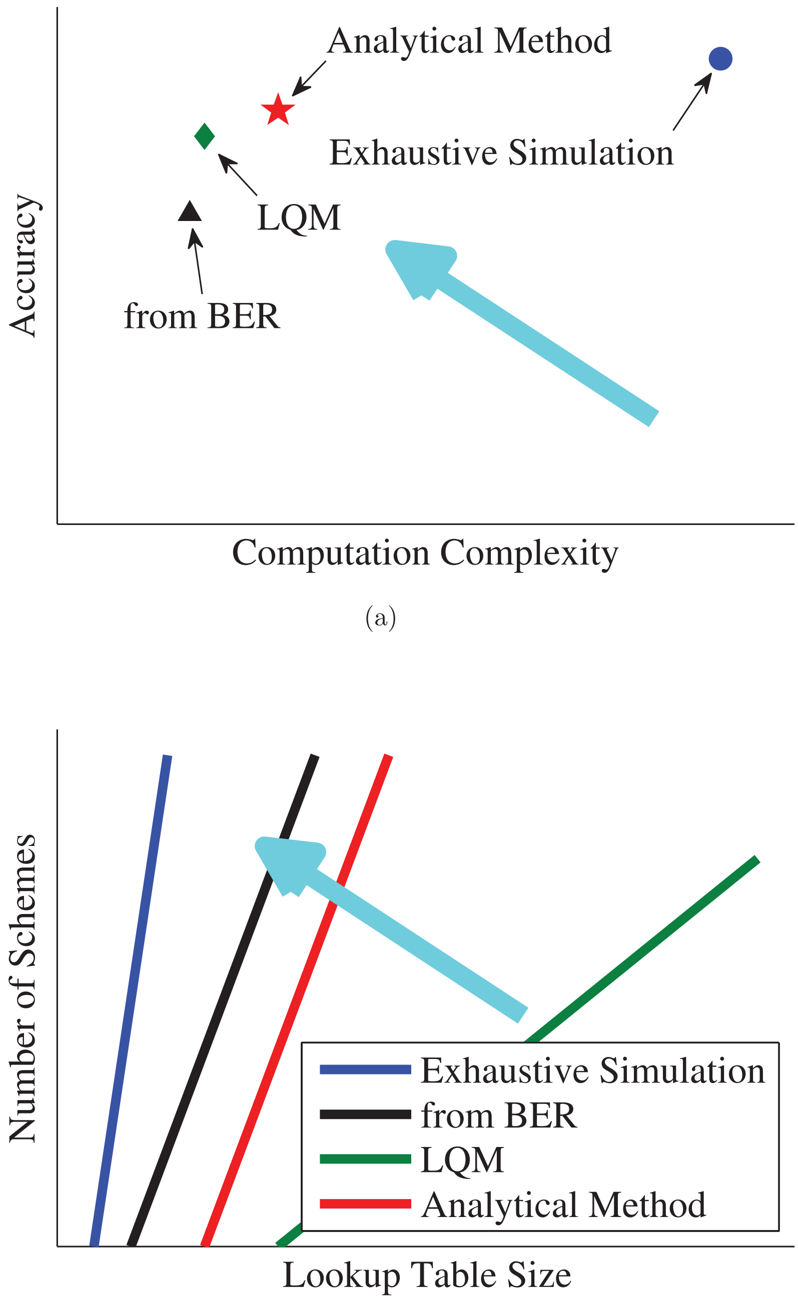
packet, and $\beta_{M,R,L}$ is a parameter to be calibrated. Note that different LQMs have different g(·). For example, the function g(·) of EESM and MIESM are an exponential function and a mutual information function, respectively. Once the scalar metric LQM is found, the PER can be obtained through a table lookup

$$\mathrm{PER}_{M,R,L} = \mathrm{f}_{M,R,L}(\mathrm{LQM})\,, \tag{1.2}$$

where $\mathrm{PER}_{M,R,L}$ is the PER of the transmission scheme specified by $\{M, R, L\}$, and $\mathrm{f}_{M,R,L}(\cdot)$ is the PER curve generated by simulations in an AWGN channel. For example, Fig. 1.2 shows the PER curves of various transmission schemes as a function of the effective SNR. Both EESM and MIESM use these curves as the mapping functions $\mathrm{f}_{M,R,L}(\cdot)$. Other methods have different mapping functions that take their own LQMs as the input. Note that these pre-simulated curves $\mathrm{f}_{M,R,L}(\cdot)$ need to be prepared for all available transmission schemes. Furthermore, the calibrated parameter $\beta_{M,R,L}$ is chosen such that it minimizes the mean square error (MSE) between the predicted PER and the pre-simulated PER for all targeted types of fading channels. Hence, a large lookup table and a considerable amount of pre-simulations are inevitable in order to support systems with numerous modes.

If computing capability allows, an exhaustive simulation is a valid way to predict the PER. This method is very accurate but has high computation complexity. It does not require any pre-simulation results in order to predict the PER of a transmission scheme, but only need to know the implementation of the scheme. If we were able to predict a BPSK rate-1/2 scheme, we only need to know the implementation of the code rate equal to 2/3 in order to predict a BPSK rate-2/3 scheme. Another method is to predict the PER from the coded BER with the assumption of the independence of bit errors. This method is very easy because the coded BER can be a simple mapping from the uncoded BER [PZR07], but it has poor accuracy. To support an additional transmission scheme, it might require some calibrations for the coded BER mapping coefficients, but the modulation and code rate are independently considered in this method. The LQMs are slightly more complicated but have a lot better prediction accuracy than the method relying on the coded BER. However, each transmission scheme is treated separately and requires its own curve-fitting and parameter calibrations. For example, being able to predict a BPSK rate-1/2 scheme does not make it easier to predict a BPSK rate-2/3 scheme.

Fig. 1.3 summarizes the PER prediction methods mentioned above, where the cyan arrows show the trend of a good method. We seek an analytical method that is more accurate than the LQMs and requires affordable computation complexity. Moreover, we would like the analytical method not to treat each transmission scheme separately but reuse some intermediate results if two schemes have some settings in common. For example, the prediction of a BPSK rate-1/2 scheme uses the same LLR distribution as the prediction of a BPSK rate-2/3 scheme.



(a)

(b)

Figure 1.3: The comparison of PER prediction methods in terms of their efficiency (a) and applicability (b).

1.1.2 Cyclic Redundancy Check Code Design

Error-detecting codes and error-correcting codes work together to guarantee a reliable link. The inner error-*correcting* code tries to correct any errors caused by the channel. If the outer error-*detecting* code detects any residual errors, then the receiver will declare a failed transmission.

Undetected errors result when an erroneously decoded codeword of the inner code has a message that is a valid codeword of the outer code. We propose a design of cyclic redundancy check (CRC) codes for a given feedforward convolutional code such that the undetected error probability is minimized.

A necessary condition for a good joint design of error-detecting and error-correcting codes, both using linear block codes, is provided in [KM84]. However, this condition is based on the minimum distances of the inner and outer codes and does not consider the detailed code structure.

Most prior work on CRC design ignores the inner code structure by assuming that the CRC code is essentially operating on a binary symmetric channel (BSC). We refer to this as the BSC assumption. The BSC assumption does not take advantage of the fact that the CRC code will only encounter error sequences that are valid codewords of the inner code.

The undetected error probability of a CRC code under the BSC assumption was evaluated using the weight enumerator of its dual code in [LBF79]. Fast algorithms to calculate dominant weight spectrum and undetected error probability of CRC codes under the BSC assumption were presented in [Kaz01, LC05].

In [KC04], Koopman and Chakravarty list all standard and good (under the BSC assumption) CRC codes with up to 16 parity bits for information lengths up to 2048 bits. The authors recommend CRC codes given the specific target redundancy length and information length.

For more than 16 CRC parity bits, it is difficult to search all possibilities and find the best CRC codes even under the BSC assumption. Some classes of CRC codes with 24 and 32 bits were investigated under the BSC assumption in [CBH93] and an exhaustive search for 32-bit CRC codes under the BSC assumption was performed later in [Koo02].

Because all of these designs ignore the inner code by assuming a BSC, there is no guarantee of optimality when these CRC codes are used with a specific inner code. Therefore, we seek an optimal design of the CRC code for a specific convolutional code such that the undetected error probability is minimized.

A few papers do consider CRC and convolutional codes together, but do not consider the undetected error probability. In [SCH07], the CRC code was jointly decoded with the convolutional code and used to detect the message length without much degradation of its error detection capability. In [GL13], CRC bits were punctured to reach higher code rates. The authors noticed that bursty bit errors caused an impact on the performance of the punctured CRC code.

1.2 Summary of Contributions

In Chapter 2, We first present the analytical PER prediction of punctured convolutionally coded systems employing a uniform random interleaver. Similar to the LQMs, the proposed PER prediction takes the PPSNR as its input. Given the PPSNR and the modulation scheme, we derive the LLR distribution. Different from the LLR approximations in [ASF09] called consistent model (CoM) and zero-crossing model (ZcM), we approximate each piecewise normal probability density function (PDF) using a single normal PDF such that the tail probability of the sum of the piecewise normal random variables approximates the tail probability of the sum of the normal random variables. Having derived the LLR distribution, we then use the Gaussian Q-function approximation [CDS03] and the transfer functions of the punctured convolutional code to calculate the sum of pairwise error probabilities (PEP) and therefore the PER. This analytical method is refined from[LD12] and delivers higher accuracy by allowing multiple transfer functions for each punctured convolutional code and a more accurately modeling of the LLR distribution for high order QAM. To compute the transfer functions efficiently, we improved the condensed state diagram [HB89, LK04] of a punctured convolutinal code to derive the exact transfer functions while keeping the number of states equal to that of the mother convolutional code. This method is referred to as the Q-function approximation method.

We also present another PER prediction method based on the saddlepoint approximation [MFC06]. Different from [KL10], which predicts BER by considering only the nearest competitive constellation point, we predict PER by taking all constellation points into account and approximating the LLR using folded normal distributions [LNN61] for high order QAM instead of normal distributions. This method is called the saddlepoint approximation method.

The proposed analytical PER prediction methods deliver higher accuracy than the existing heuristic LQMs while requiring absolutely no curve-fitting or parameter calibrations. In our simulations in Rayleigh fading channels, the best heuristic method has averaged relative prediction error of 8.07%, while the prediction errors of the Q-function approximation and the saddlepoint approximation methods are 4.60% and 6.69%, respectively. In the 2x2 frequency selective channel settings, the averaged MSE of the best heuristic method is 0.0504, while the Q-function approximation and the saddlepoint approximation methods have MSEs of 0.0264 and 0.0236, respectively.

In Chapter 3, We present the analytical PER prediction for punctured convolutionally coded MIMO OFDM systems employing a repeated-pattern interleaver. The correlation of LLRs brought about by the interleaver are discussed under three categories: a) correlation across subcarriers, b) correlation across spatial streams, and c) correlation within a subcarrier. The location of a bit determines the effective channel it experiences, and this determines its LLR distribution. If two bits belong to the same subcarrier but different spatial streams, they may be influenced by the same noise source but with different weightings. If two bits belong to the same subcarrier and the same spatial stream, they may observe exactly the same noise realization and have highly correlated LLRs. These are all considered within the framework of the Q-function approximation method. This method achieves about 10% of the averaged MSE compared to the other methods, which assume the system is using a uniform random interleaver. Again, the Q-function approximation method is based on the analysis of the error events and requires no simulations, curve-fitting, or parameter calibrations.

Motivated by the analytical PER prediction, we apply the concept to the calculation of the undetected error probability of a CRC code concatenated with a feedforward convolutional code and propose two methods in Chapter 4. The *exclusion* method enumerates possible error patterns of the inner code and excludes them one by one if they are detectable. The *construction* method constructs a new convolutional code whose error events correspond exactly to the undetectable error events of the original concatenation of CRC and convolutional codes. Therefore, the undetected error probability is approximated as the PER of the new convolutional code. The new convolutional code is a catastrophic convolutional code and its generator polynomial is the product of the CRC generator polynomial and the original convolutional generator polynomial. With these two methods as tools, we design CRC codes for the most common 64-state convolutional code for information length $k = 1024$ bits and compare with existing CRC codes, demonstrating the performance benefits of utilizing the inner code structure. Under this setting, the new CRC codes provide significant reduction in undetected error probability compared to the existing CRC codes with the same degrees. With the proposed design, we are able to save two check bits in most cases while having the same error detection capability.

CHAPTER 2

Analytical Packet Error Rate Predictions of Punctured Convolutional Codes with Bit-Interleaved Coded Modulation

Existing packet error rate (PER) prediction methods such as exponential effective SNR mapping (EESM) and mean mutual information per bit (MMIB) require off-line simulations, curve-fitting, and parameter calibrations for every combination of modulation, code rate, and packet length. These requirements lower their applicability as modern systems support numerous modes (combinations of the above) thus needing an impractical number of offline simulations. We propose two analytical PER prediction methods for punctured convolutional codes based on the codes' transfer functions and the modeling of the log-likelihood ratio (LLR) distributions. The first method approximates the LLR using a mixture of normal distributions and relies on the Gaussian Q-function approximation; the second method models the LLR as a mixture of folded normal distributions and applies the saddlepoint approximation. Both methods are quite versatile and deliver high accuracy for arbitrary combinations of modulation schemes, code rates, packet size, channels, etc. They are equally applicable to a wide range of systems from single-input single-output (SISO) single carrier to multiple-input multiple-output (MIMO) orthogonal frequency division multiplexing (OFDM).

This chapter is organized as follows: Section 2.1 provides the system model. In Section 2.2, we derive the PER of a punctured convolutional code as a function

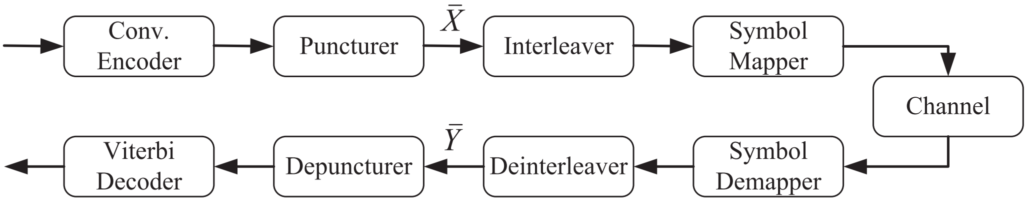


Figure 2.1: The block diagram of a system employing punctured convolutional codes with BICM.

of the pairwise error probabilities (PEP). The two proposed PEP calculations are described in Section 2.3. In Section 2.4, the numerical verification and comparison is presented. Finally, we conclude this chapter in Section 2.5.

Notation: Boldface letters **x** represent sets. Vectors and matrices are denoted by letters with a bar $\bar{x}$. Random variables are written as italic uppercase letters X, and their realizations are represented by the corresponding italic lowercase letters x.

2.1 System Model

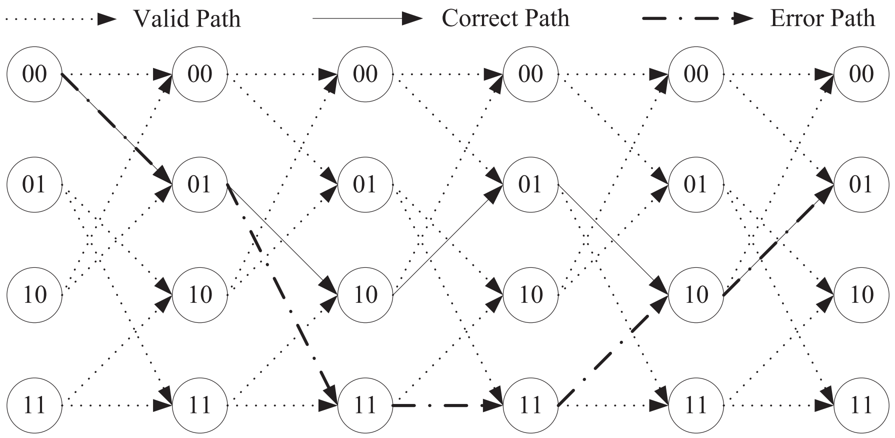
Fig. 2.1 illustrates a system employing punctured convolutional codes with bit-interleaved coded modulation (BICM). In this system, a length-L information bit sequence, constituting a packet, is fed into a convolutional encoder, and the coded bit sequence can be punctured to reach a higher code rate. The punctured bit sequence $\bar{X}$ is then interleaved by a uniform random interleaver, which generates each possible permutation of the sequence with equal probability. Afterward the interleaved bit sequence is modulated using binary phase-shift keying (BPSK), quadrature phase-shift keying (QPSK), 16-quadrature amplitude modulation (QAM), or 64-QAM with Gray code, and the mapping rule follows [Sta09]. We put no constraint on the antenna configurations; the system can use either single antenna or multiple antennas with MIMO techniques, e.g. beamforming, spatial multiplexing, space-time code. However, we do assume that the post-processing signal-to-noise ratio (PPSNR)[HSP01], which is defined as the effective signal-to-noise ratio (SNR) right before the symbol demapper, of each modulated symbol is known and the noise follows an independent zero-mean circularly-symmetric complex normal distribution. The receiver is equipped with a simplified soft-demapper[TB02], which is an approximation of the max-log demapper[Vit98] but requires rather simple computations. Finally, the deinterleaved LLR sequence $\bar{Y}$ is depunctured and decoded by a soft decision Viterbi decoder.

Note, because of the uniform random interleaver, the punctured bits are mapped to each modulation symbol with equal probability and therefore the PPSNRs associated to all punctured bits can be treated as identically distributed random variables. For ease of analysis, we further assume that these PPSNRs are independent.

2.2 PER of Punctured Convolutional Codes

In the analysis, we treat the convolutional code encoder and the puncturer as one combined encoder. This combined encoder is still analyzable using its trellis diagram with the total number of states increased to the product of the original number of encoder states and the number of puncture states. For example, a 64-state convolutional code encoder combining with a 3-state puncturer will yield a 192-state combined encoder.

In a trellis diagram, an error event occurs if the decoded trellis path deviates from the correct path exactly once and rejoins it exactly once. Let $\boldsymbol{\varepsilon}_{t,l}$ be the error event whose deviation starts at the l^{th} state transition (or information bit) and the deviant trajectory is indexed as t. Let $\bar{\varepsilon}_{t,l}$ be the bit sequence generated by bitwise exclusive-or (XOR) of: (a) the encoder output along the correct trellis path $\bar{X}$ with (b) the encoder output along the error trellis path of error event $\boldsymbol{\varepsilon}_{t,l}$.



Correct Path Input	1	0	1	0	1
Correct Path Output	11	10	00	10	00
Error Path Input	1	1	1	0	1
Error Path Output	11	01	10	01	00
XOR(Input)	0	1	0	0	0
XOR(Output)	00	11	10	11	00

Figure 2.2: An error event $\boldsymbol{\varepsilon}_{1,2}$ shown in the trellis diagram of a $[7,5]_8$ convolutional code with no puncture.

Note that the error trajectory index t can be determined by the bitwise XOR of the input sequences along the two paths. For example, in Fig. 2.2, the error path leaves the correct path at the second state transition, so $l = 2$. If the trajectory is indexed as $t = 1$, then $\bar{\varepsilon}_{1,2} = [0011101100\cdots]$. For each error event $\boldsymbol{\varepsilon}_{t,l}$, define pairwise error event $\mathbf{e}_{t,l}$, which occurs when the received LLR sequence $\bar{Y}$ is more likely to follow the error trellis path of $\bar{X} \oplus \bar{\varepsilon}_{t,l}$ than to follow the correct path of $\bar{X}$, where $\oplus$ is the bitwise XOR operator.

A packet error happens when at least one error event occurs in this packet. Therefore, the PER expression for a length-L packet is simply given as

$$\mathrm{PER}_L = \mathrm{P}\left(\bigcup_{l=1}^{L} \bigcup_{t \in \mathbf{T}_{L-l+1}} \boldsymbol{\varepsilon}_{t,l}\right) = \mathrm{P}\left(\bigcup_{l=1}^{L} \bigcup_{t \in \mathbf{T}_{L-l+1}} \mathbf{e}_{t,l}\right), \tag{2.1}$$

where the finite set $\mathbf{T}_l$ is a collection of indices of the error trellis paths whose deviant sections contain at most l state transitions. Moreover, because of the time-invariance of convolutional codes, $\mathbf{T}_{L-l+1}$ actually represents the set of indices of error trellis paths starting at the l^{th} information bit in a length-L packet. Assume that $\bigcup_{t \in \mathbf{T}_{L-l+1}} \mathbf{e}_{t,l}, l = 1, 2, \ldots, L$ are independent and apply the union bound. Then the PER can be approximated by

$$\mathrm{PER}_L \approx 1 - \prod_{l=1}^{L} \left[1 - \mathrm{P}\left(\bigcup_{t \in \mathbf{T}_{L-l+1}} \mathbf{e}_{t,l} \right) \right] \leq 1 - \prod_{l=1}^{L} \left[1 - \sum_{t \in \mathbf{T}_{L-l+1}} \mathrm{P}(\mathbf{e}_{t,l}) \right]. \quad (2.2)$$

Define the infinite set $\mathbf{T}$ as the collection of indices of all error trellis paths without the length limitation. The relationship between $\mathbf{T}$ and $\mathbf{T}_l$ is

$$\mathbf{T}_1 \subseteq \mathbf{T}_2 \subseteq \cdots \subseteq \mathbf{T}_{L-1} \subseteq \mathbf{T}_L \subset \mathbf{T}.$$

Therefore, by replacing $\mathbf{T}_{L-l+1}$ with $\mathbf{T}$, the upper bound of (2.2) is obtained.

Let M be the number of puncture states. Since convolutional code encoders are time-invariant and the puncture pattern is applied periodically, $\bar{\varepsilon}_{t,l}$ and $\bar{\varepsilon}_{t,l+M}$ have equal Hamming weights for all t and l. Thus the PER of a length-L packet is approximated by

$$\mathrm{PER}_L \approx 1 - \prod_{l=1}^{M} \left[1 - \sum_{t \in \mathbf{T}} \mathrm{P}(\mathbf{e}_{t,l}) \right]^{L/M}. \quad (2.3)$$

When no puncturing is applied, we have $M = 1$, and (2.3) can be written as

$$\mathrm{PER}_L \approx 1 - \left[1 - \sum_{t \in \mathbf{T}} \mathrm{P}(\mathbf{e}_t) \right]^{L}, \quad (2.4)$$

where $\mathbf{e}_t$ is the pairwise error event regardless of its starting location. This formula, first seen in[PT87], was proven as a tight PER upper bound for hard-decision Viterbi decoding, assuming independent and identically distributed (i.i.d.) bit errors at the decoder input. For soft-decisions, it was first claimed to be an upper bound in [KL92]. However, in its proof, the probability of choosing a correct branch in the decoder trellis should not be lower bounded by $1 - \sum_{t \in \mathbf{T}} \mathrm{P}(\mathbf{e}_t)$, so we claim that (2.4) is only an approximation.

It is worth noting that in low SNR regions, the union bound could make $\sum_{t \in \mathbf{T}} \mathrm{P}(\mathbf{e}_{t,l})$ in (2.3) exceed one. In such cases, we set the summation equal to

Table 2.1: The scaled in-phase component of a received symbol

Modulation	Z_{I}	g
BPSK	$\mathrm{sgn}(X_{\mathrm{I},1}) + n_{\mathrm{I}}\sqrt{2g/E_{\mathrm{s}}}$	0.5
QPSK	$\mathrm{sgn}(X_{\mathrm{I},1}) + n_{\mathrm{I}}\sqrt{2g/E_{\mathrm{s}}}$	1
16-QAM	$\mathrm{sgn}(X_{\mathrm{I},1})[2 - \mathrm{sgn}(X_{\mathrm{I},2})] + n_{\mathrm{I}}\sqrt{2g/E_{\mathrm{s}}}$	5
64-QAM	$\mathrm{sgn}(X_{\mathrm{I},1})\{4 - \mathrm{sgn}(X_{\mathrm{I},2})[2 - \mathrm{sgn}(X_{\mathrm{I},3})]\} + n_{\mathrm{I}}\sqrt{2g/E_{\mathrm{s}}}$	21

one in order to avoid unreasonable result. Unlike the limiting before averaging technique in [ML99], this limiting is performed after considering the variation of the channel in $\mathrm{P}(\mathbf{e}_{t,l})$.

In the following section, we will focus on calculating the PEP $\mathrm{P}(\mathbf{e}_{t,l})$ for different modulation schemes and arbitrary fading coefficients.

2.3 PEP Calculation

To calculate the PEP, we need to know the distribution of the LLRs associated with each error event. For simplicity, let Z_{I} be the in-phase component of one received QAM symbol scaled by $\sqrt{2g/E_{\mathrm{s}}}$, where g is a modulation dependent scaling factor and E_{s} is the received symbol energy. The expressions of Z_{I} and the values of g are given in Table 2.1. In this table, the use of Gray code is assumed, $\mathrm{sgn}(\cdot)$ is a function which maps 1 to 1 and 0 to -1, and $X_{\mathrm{I},k}$ denotes the k^{th} bit of the in-phase component of the symbol, e.g. $X_{\mathrm{I},1}$ for the most significant bit (MSB). Furthermore, n_{I} is the in-phase component of the white Gaussian noise, which has zero mean and variance σ^2. Table 2.2 summarizes the results of [TB02] and gives the in-phase LLRs calculated by the simplified demapper. In Table 2.2, $Y_{\mathrm{I},k}$ is the LLR of $X_{\mathrm{I},k}$, and $\mathit{\Gamma} = \frac{E_{\mathrm{s}}}{2\sigma^2}$ is the PPSNR of the symbol. The quadrature components are processed in a similar manner.

Recall that $\bar{X}$ is the bit sequence after the puncturer, and $\bar{Y}$ is the LLR

Table 2.2: The LLRs of the in-phase component calculated by the simplified demapper[TB02].

Modulation	$Y_{\mathrm{I},1}$	$Y_{\mathrm{I},2}$	$Y_{\mathrm{I},3}$
BPSK	$-2Z_{\mathrm{I}}\mathit{\Gamma}/g$	N/A	N/A
QPSK	$-2Z_{\mathrm{I}}\mathit{\Gamma}/g$	N/A	N/A
16-QAM	$-2Z_{\mathrm{I}}\mathit{\Gamma}/g$	$-2\left(-\lvert Z_{\mathrm{I}}\rvert+2\right)\mathit{\Gamma}/g$	N/A
64-QAM	$-2Z_{\mathrm{I}}\mathit{\Gamma}/g$	$-2\left(-\lvert Z_{\mathrm{I}}\rvert+4\right)\mathit{\Gamma}/g$	$-2\big(-\big\lvert\lvert Z_{\mathrm{I}}\rvert-4\big\rvert+2\big)\mathit{\Gamma}/g$

sequence before the depuncturer. The i^{th} element of $\bar{Y}$, $\bar{\varepsilon}_{t,l}$, and $\bar{X}$ are denoted by Y_i, $\varepsilon_{t,l,i}$, and X_i, respectively. Given a modulation scheme, Y_i is calculated according to either of the columns in Table 2.2 with equal probability because of the uniform random interleaver. With $\bar{X}$ and $\bar{Y}$, we express the PEP $\mathrm{P}(\mathbf{e}_{t,l})$ as

$$\mathrm{P}(\mathbf{e}_{t,l}) = \mathrm{P}\big(\bar{X} \to \bar{X} \oplus \bar{\varepsilon}_{t,l}\big) = \mathrm{P}\left(\sum_{\varepsilon_{t,l,i}\neq 0} Y_i \,\mathrm{sgn}(X_i) > 0\right). \tag{2.5}$$

For simplicity, define half of log-error-likelihood ratio (HLELR) of Y_i as

$$Y_i^{\star} \triangleq Y_i \,\mathrm{sgn}(X_i)\,/\,2\,. \tag{2.6}$$

Thus the PEP is now given by

$$\mathrm{P}(\mathbf{e}_{t,l}) = \mathrm{P}\left(\sum_{\varepsilon_{t,l,i}\neq 0} Y_i^{\star} > 0\right) = \mathrm{P}\left(\sum_{j=1}^{w_{t,l}} Y^{\star}_{\mathrm{i}(t,l,j)} > 0\right), \tag{2.7}$$

where $w_{t,l}$ is the Hamming weight of $\bar{\varepsilon}_{t,l}$ and the function $\mathrm{i}(t,l,j)$ outputs the location of the j^{th} nonzero bit of $\bar{\varepsilon}_{t,l}$. To calculate this probability, we need to know the distribution of the HLELR. Consider BPSK and QPSK. Using Table 2.1, Table 2.2, (2.6), and the fact that the noise is independent of the transmitted bit X_i, we obtain the conditional distribution

$$Y_i^{\star}|(\mathit{\Gamma}_i = \gamma_i) \sim \mathcal{N}(\,-\gamma_i/g,\, \gamma_i/g\,)\,,$$

where $\mathit{\Gamma}_i$ is the PPSNR of the symbol associated to the i^{th} bit. Note that, this likelihood ratio expresses how likely it is for the bit to be in error, so its expectation is always negative.

For higher order QAMs, the HLELR distribution depends on the bit's significance in the modulation symbol and should be discussed separately. If the realizations of $\mathit{\Gamma}$ and $X_{\mathrm{I},k}$ for all k are given, the conditional distribution of Z_{I} is normal, and thus the conditional distribution of the MSB HLELR $Y^{\star}_{\mathrm{I},1}$ is also normal. The conditional variance of $Y^{\star}_{\mathrm{I},1}$ given $\mathit{\Gamma} = \gamma$ and $X_{\mathrm{I},k} = x_{\mathrm{I},k}$ for all k is $\frac{\gamma}{g}$ and its conditional expectation is

$$\mathrm{E}\big[Y^{\star}_{\mathrm{I},1}\big|\mathit{\Gamma} = \gamma, X_{\mathrm{I},k} = x_{\mathrm{I},k}\,\forall k\big] = \begin{cases} -\gamma/g & \text{16-QAM}, x_{\mathrm{I},2} = 1 \\ -3\gamma/g & \text{16-QAM}, x_{\mathrm{I},2} = 0 \\ -\gamma/g & \text{64-QAM}, x_{\mathrm{I},2} = 1, x_{\mathrm{I},3} = 0 \\ -3\gamma/g & \text{64-QAM}, x_{\mathrm{I},2} = 1, x_{\mathrm{I},3} = 1 \\ -5\gamma/g & \text{64-QAM}, x_{\mathrm{I},2} = 0, x_{\mathrm{I},3} = 1 \\ -7\gamma/g & \text{64-QAM}, x_{\mathrm{I},2} = 0, x_{\mathrm{I},3} = 0 \end{cases}.$$

To simplify the expression, we introduce another random variable $D_{\mathrm{I},1}$, which is governed by $X_{\mathrm{I},2}$ and $X_{\mathrm{I},3}$, so that the conditional distribution of $Y^{\star}_{\mathrm{I},1}$ can be written as

$$Y^{\star}_{\mathrm{I},1}\big|(D_{\mathrm{I},1} = d_{\mathrm{I},1}, \mathit{\Gamma} = \gamma) \sim \mathcal{N}\big(-d_{\mathrm{I},1}\gamma/g\,, \gamma/g\big)\,. \tag{2.8}$$

By letting $D_{\mathrm{I},1} = 1$ for BPSK and QPSK, (2.8) can be applied to all modulation schemes.

During the computation of $Y_{\mathrm{I},2}$ in Table 2.2, we need to take the absolute value of Z_{I}. Since the conditional distribution of Z_{I} given $\mathit{\Gamma}$ and $X_{\mathrm{I},k}$ for all k is normal, its absolute value makes the conditional distributions of $Y_{\mathrm{I},2}$ and $Y^{\star}_{\mathrm{I},2}$ piecewise normal, or more specifically, folded normal [LNN61]. Consider a normal distribution with mean μ and variance ρ^2 folded at $\mu - \eta$ to its center, where

$\eta \neq 0$, i.e. folded to the right if $\eta > 0$ and vice versa. We denote this distribution as $\mathcal{FN}(\mu, \rho^2, \eta)$. Its probability density function (PDF) is

$$f_{\mathrm{FN}}(x) = \frac{1}{\sqrt{2\pi\rho^2}} \left[e^{\frac{(x-\mu)^2}{-2\rho^2}} + e^{\frac{(x-\mu+2\eta)^2}{-2\rho^2}} \right] \mathrm{I}\left(\frac{x}{\eta} > \frac{\mu - \eta}{\eta} \right), \tag{2.9}$$

where $\mathrm{I}(\cdot)$ is the indicator function. The presence of η in the denominator in the indicator function makes this expression valid for both positive and negative η. In the case of $Y^{\star}_{\mathrm{I},2}$, we have $\mu = \frac{d_{\mathrm{I},2}\gamma}{-g}$, $\eta = \frac{\phi_{\mathrm{I},2}\gamma}{g}$, and $\rho^2 = \frac{\gamma}{g}$, where

$$(d_{\mathrm{I},2}, \phi_{\mathrm{I},2}) = \begin{cases} (1,1) & \text{16-QAM}, x_{\mathrm{I},2} = 1 \\ (1,-3) & \text{16-QAM}, x_{\mathrm{I},2} = 0 \\ (3,1) & \text{64-QAM}, x_{\mathrm{I},2} = 1, x_{\mathrm{I},3} = 0 \\ (1,3) & \text{64-QAM}, x_{\mathrm{I},2} = 1, x_{\mathrm{I},3} = 1 \\ (1,-5) & \text{64-QAM}, x_{\mathrm{I},2} = 0, x_{\mathrm{I},3} = 1 \\ (3,-7) & \text{64-QAM}, x_{\mathrm{I},2} = 0, x_{\mathrm{I},3} = 0 \end{cases}.$$

To simplify the expression, two random variables $D_{\mathrm{I},2}$ and $\mathit{\Phi}_{\mathrm{I},2}$ are introduced. The random variable $D_{\mathrm{I},2}$, whose realization is $d_{\mathrm{I},2}$, is governed by $X_{\mathrm{I},3}$; the random variable $\mathit{\Phi}_{\mathrm{I},2}$, whose realization is $\phi_{\mathrm{I},2}$, is governed by $X_{\mathrm{I},2}$ and $X_{\mathrm{I},3}$. Therefore, the conditional distribution of $Y^{\star}_{\mathrm{I},2}$ can be written as

$$Y^{\star}_{\mathrm{I},2} \big| (D_{\mathrm{I},2} = d_{\mathrm{I},2}, \mathit{\Phi}_{\mathrm{I},2} = \phi_{\mathrm{I},2}, \mathit{\Gamma} = \gamma) \sim \mathcal{FN}(-d_{\mathrm{I},2}\gamma/g\,,\, \gamma/g\,,\, \phi_{\mathrm{I},2}\gamma/g)\,. \tag{2.10}$$

To compute $Y_{\mathrm{I},3}$ for 64-QAM, we need to take the absolute value of Z_{I} twice as shown in Table 2.2. Because of this, the conditional distribution of $Y^{\star}_{\mathrm{I},3}$ given $\mathit{\Gamma}$ and $X_{\mathrm{I},k}$ for all k is folded normal with three folds. For simplicity, we only consider the fold closest to its center, i.e. the one with the smallest $|\eta|$, because it has the greatest impact. Therefore, for $Y^{\star}_{\mathrm{I},3}$, we have $\mu = \frac{\gamma}{-g}$, $\eta = \frac{\phi_{\mathrm{I},3}\gamma}{g}$, and

Table 2.3: The possible realization pairs of $(D_{\mathrm{I},k}, \Phi_{\mathrm{I},k})$.

Modulation	$(d_{\mathrm{I},1}, \phi_{\mathrm{I},1})$	$(d_{\mathrm{I},2}, \phi_{\mathrm{I},2})$	$(d_{\mathrm{I},3}, \phi_{\mathrm{I},3})$
BPSK	$(1, \infty)$	N/A	N/A
QPSK	$(1, \infty)$	N/A	N/A
16-QAM	$(1, \infty), (3, \infty)$	$(1, 1), (1, -3)$	N/A
64-QAM	$(1, \infty), (3, \infty),$ $(5, \infty), (7, \infty)$	$(3, 1), (1, 3),$ $(1, -5), (3, -7)$	$(1, 1), (1, 1),$ $(1, 1), (1, -3)$

$\rho^2 = \frac{\gamma}{g}$, where

$$\phi_{\mathrm{I},3} = \begin{cases} 1 & x_{\mathrm{I},2} = 1, x_{\mathrm{I},3} = 0 \\ 1 & x_{\mathrm{I},2} = 1, x_{\mathrm{I},3} = 1 \\ 1 & x_{\mathrm{I},2} = 0, x_{\mathrm{I},3} = 1 \\ -3 & x_{\mathrm{I},2} = 0, x_{\mathrm{I},3} = 0 \end{cases}.$$

Similarly, we introduce the random variable $\Phi_{\mathrm{I},3}$, whose realization is $\phi_{\mathrm{I},3}$, governed by $X_{\mathrm{I},2}$ and $X_{\mathrm{I},3}$. The conditional distribution is then given by

$$Y^{\star}_{\mathrm{I},3}\big|(\Phi_{\mathrm{I},3} = \phi_{\mathrm{I},3}, \Gamma = \gamma) \sim \mathcal{FN}(-\gamma/g\,,\, \gamma/g\,,\, \phi_{\mathrm{I},3}\gamma/g)\,. \tag{2.11}$$

To further simplify the expression of HLELRs, we introduce random variables $\Phi_{\mathrm{I},1}$ and $D_{\mathrm{I},3}$ so that

$$Y^{\star}_{\mathrm{I},k}\big|(D_{\mathrm{I},k} = d_{\mathrm{I},k}, \Phi_{\mathrm{I},k} = \phi_{\mathrm{I},k}, \Gamma = \gamma) \sim \mathcal{FN}(-d_{\mathrm{I},k}\gamma/g\,,\, \gamma/g\,,\, \phi_{\mathrm{I},k}\gamma/g) \tag{2.12}$$

is valid for all k. Note that a folded normal distribution with its fold at infinity is actually a normal distribution. The possible realization pairs of $(D_{\mathrm{I},k}, \Phi_{\mathrm{I},k})$ for all k are listed in Table 2.3. Moreover, since the punctured bits $X_{\mathrm{I},2}$ and $X_{\mathrm{I},3}$ in a QAM symbol are treated as i.i.d. Bernoulli random variables with parameter 0.5, all realization pairs of each modulation are equally probable. Because of the uniform random interleaver, a punctured bit can be any significant bit in a symbol with equal probability. Therefore, by introducing random variable pair (D_i, Φ_i), which is a mixture of $(D_{\mathrm{I},k}, \mathit{\Phi}_{\mathrm{I},k})$ for all k with equal weights, we can express the conditional distribution of an HLELR as

$$Y_i^\star | (D_i = d_i, \mathit{\Phi}_i = \phi_i, \mathit{\Gamma}_i = \gamma_i) \sim \mathcal{FN}\left(\frac{d_i\gamma_i}{-g}, \frac{\gamma_i}{g}, \frac{\phi_i\gamma_i}{g}\right). \tag{2.13}$$

In the following, we will present two methods to compute the PEP in (2.7) based on (2.13).

2.3.1 PEP Based on Gaussian Q-Function Approximation

To simplify the computation of the tail probability of the HLELR sum in (2.7), the folded normal distribution in (2.13) is approximated by a single normal distribution. Different from the two single normal approximations proposed in [ASF09], we seek $Y_i^{\star\star}$, an approximation of $Y_i^\star$, such that

$$\begin{aligned} &\mathrm{P}\left(\sum_{j=1}^{w_{t,l}} Y^\star_{\mathrm{i}(t,l,j)} > 0 \,\middle|\, \begin{array}{l} D_i = d_i, \mathit{\Phi}_i = \phi_i, \\ \mathit{\Gamma}_i = \gamma_i, i = \mathrm{i}(t,l,j') \end{array}\right) \\ &\approx \mathrm{P}\left(Y_i^{\star\star} + \sum_{j=1, j\neq j'}^{w_{t,l}} Y^\star_{\mathrm{i}(t,l,j)} > 0 \,\middle|\, \begin{array}{l} D_i = d_i, \mathit{\Phi}_i = \phi_i, \\ \mathit{\Gamma}_i = \gamma_i, i = \mathrm{i}(t,l,j') \end{array}\right). \end{aligned}$$

Although it is possible to obtain the approximation for each $w_{t,l} \geq d_{\mathrm{free}}$, where d_{free} is the free distance of the punctured convolutional code, we consider the dominant case $w_{t,l} = d_{\mathrm{free}}$ and apply the approximation to all $w_{t,l} \geq d_{\mathrm{free}}$. To be more specific, $Y_i^\star$ is approximated by $Y_i^{\star\star}$ such that

$$\begin{aligned} &\mathrm{P}\left(\sum_{j=1}^{d_{\mathrm{free}}} Y^\star_{\mathrm{i}(t,l,j)} > 0 \,\middle|\, \begin{array}{l} D_i = d_i, \mathit{\Phi}_i = \phi_i, \\ \mathit{\Gamma}_i = \gamma_i, i = \mathrm{i}(t,l,j') \end{array}\right) \\ &\approx \mathrm{P}\left(Y_i^{\star\star} + \sum_{j=1, j\neq j'}^{d_{\mathrm{free}}} Y^\star_{\mathrm{i}(t,l,j)} > 0 \,\middle|\, \begin{array}{l} D_i = d_i, \mathit{\Phi}_i = \phi_i, \\ \mathit{\Gamma}_i = \gamma_i, i = \mathrm{i}(t,l,j') \end{array}\right), \end{aligned} \tag{2.14}$$

and

$$Y_i^{\star\star} | (D_i = d_i, \mathit{\Phi}_i = \phi_i, \mathit{\Gamma}_i = \gamma_i) \sim \mathcal{N}\left(-\left[d_i + \mathrm{d_{adj}}\left(\frac{d_i\gamma_i}{-g}, \frac{\gamma_i}{g}, \frac{\phi_i\gamma_i}{g}\right)\right]\frac{\gamma_i}{g}, \frac{\gamma_i}{g}\right). \tag{2.15}$$

The adjustment function $\mathrm{d_{adj}}(\cdot)$ can be treated as a correction for the folded normal distribution and is given in Appendix 2.A.

Since $Y^{\star\star}_{\mathrm{i}(t,l,j)}$ for all j are conditionally independent given $\bar{X}$ and $\mathit{\Gamma}_{\mathrm{i}(t,l,j)}$ for all j, the conditional distribution of their sum is

$$\sum_{j=1}^{w_{t,l}} Y^{\star\star}_{\mathrm{i}(t,l,j)} \bigg| \left(D'_j = d'_j, \mathit{\Gamma}'_j = \gamma'_j, \forall j\right) \sim \mathcal{N}\left(-\sum_{j=1}^{w_{t,l}} d'_j \gamma'_j, \sum_{j=1}^{w_{t,l}} \gamma'_j\right), \tag{2.16}$$

where we define random variables

$$D'_j \triangleq D_{\mathrm{i}(t,l,j)} + \mathrm{d_{adj}}\left(\frac{D_{\mathrm{i}(t,l,j)}\mathit{\Gamma}_{\mathrm{i}(t,l,j)}}{-g}, \frac{\mathit{\Gamma}_{\mathrm{i}(t,l,j)}}{g}, \frac{\mathit{\Phi}_{\mathrm{i}(t,l,j)}\mathit{\Gamma}_{\mathrm{i}(t,l,j)}}{g}\right)$$

and $\mathit{\Gamma}'_j \triangleq \mathit{\Gamma}_{\mathrm{i}(t,l,j)} \big/ g$. The PEP is now written as

$$\begin{aligned} \mathrm{P}(\mathbf{e}_{t,l}) &\approx \mathrm{E}\left[\mathrm{P}\left(\sum_{j=1}^{w_{t,l}} Y^{\star\star}_{\mathrm{i}(t,l,j)} > 0 \bigg| D'_j, \mathit{\Gamma}'_j, \forall j\right)\right] \\ &= \mathrm{E}\left[\mathrm{Q}\left(\sqrt{\sum_{j=1}^{w_{t,l}} \mathit{\Gamma}'_j \left\{D'_j + \left(D'_j - 1\right) \frac{\sum_{j'=1}^{w_{t,l}} D'_{j'}\mathit{\Gamma}'_{j'}}{\sum_{j'=1}^{w_{t,l}} \mathit{\Gamma}'_{j'}}\right\}}\right)\right], \end{aligned} \tag{2.17}$$

where the expectation is taken over all conditioned elements, and $\mathrm{Q}(\cdot)$ is the tail probability function of the standard normal distribution, i.e. Gaussian Q-function.

To further simplify the equation, an approximation of the Gaussian Q-function is needed. Instead of the well-known Chernoff bound, which has only one exponential term, we use the sum of four exponential terms

$$\mathrm{Q}\left(\sqrt{x}\right) \approx \sum_{r=1}^{4} \alpha_r \exp(-\beta_r x), \tag{2.18}$$

where the coefficients α_r and β_r are derived according to [CDS03] and listed in Table 2.4. These numbers are selected to deliver a "close enough" approximation of the Gaussian Q-function over a wide range of inputs (one can certainly propose another set of coefficients or a formula with more terms to get a better approximation). Using (2.18),

$$\sum_{j'=1}^{w_{t,l}} D'_{j'}\mathit{\Gamma}'_{j'} \bigg/ \sum_{j'=1}^{w_{t,l}} \mathit{\Gamma}'_{j'} \approx \mathrm{E}[D'] \,,$$

Table 2.4: The coefficients of Gaussian Q-function approximation.

r	1	2	3	4
α_r	0.1453	0.1040	0.1047	0.0523
β_r	1.6212	0.7986	0.5581	0.5000

and the fact that $\left(D'_j, \mathit{\Gamma}'_j\right)$ for all j are i.i.d., we can approximate (2.17) as

$$\mathrm{P}(\mathbf{e}_{t,l}) \approx \sum_{r=1}^{4} \alpha_r \, \mathrm{E}\left[e^{-\beta_r \mathit{\Gamma}' \{D' + (D'-1)\,\mathrm{E}[D']\}}\right]^{w_{t,l}}, \tag{2.19}$$

where $(D', \mathit{\Gamma}')$ has the same distribution as $\left(D'_j, \mathit{\Gamma}'_j\right)$ without the location information. Note that the expectation now is only taken over one pair of random variables $(D', \mathit{\Gamma}')$ instead of $w_{t,l}$ pairs.

To calculate the PER, we need to sum all the PEPs starting at the same location. This task can be achieved by evaluating the transfer function, which records the encoder output weight spectrum. However, since the convolutional code encoder and the puncturer are combined, the weight spectra at different puncture states are different. Therefore, M distinct transfer functions are required. Please see Appendix 2.B for the computation of the transfer functions. Let the transfer function of the l^{th} puncture state be

$$\mathrm{T}_l(W) = \sum_{t=1}^{\infty} W^{w_{t,l}}. \tag{2.20}$$

According to (2.19) and (2.20), the sum of the PEPs in (2.3) is then approximated as

$$\sum_{t \in \mathbf{T}} \mathrm{P}(\mathbf{e}_{t,l}) \approx \sum_{r=1}^{4} \alpha_r \, \mathrm{T}_l(W)\Big|_{W=\mathrm{E}[\exp(-\beta_r \mathit{\Gamma}'\{D'+(D'-1)\,\mathrm{E}[D']\})]}. \tag{2.21}$$

Finally, the PER can be computed through (2.3) and (2.21).

2.3.2 PEP Based on Saddlepoint Approximation

In (2.7), since $Y_i^\star$ for all i are i.i.d. random variables, we apply the saddlepoint approximation [MFC06] resulting in the following expression for $\mathrm{P}(\mathbf{e}_{t,l})$

$$\mathrm{P}(\mathbf{e}_{t,l}) \approx \frac{(\mathrm{M}(\hat{s}))^{w_{t,l}+0.5}}{\hat{s}\sqrt{2\pi w_{t,l}\,\mathrm{M}''(\hat{s})}}, \tag{2.22}$$

where $\mathrm{M}(s)$ is the moment-generating function of the HLELR $Y_i^\star$. The first and second derivatives of $\mathrm{M}(s)$ are denoted as $\mathrm{M}'(s)$ and $\mathrm{M}''(s)$, respectively. The saddlepoint $\hat{s} > 0$, satisfying $\mathrm{M}'(\hat{s}) = 0$, exists and is unique [MFC06].

According to (2.13), the moment-generating function of $Y_i^\star$ is given by

$$\begin{aligned} \mathrm{M}(s) &= \mathrm{E}\big[\mathrm{E}\big(e^{sY_i^\star}\big|D_i, \varPhi_i, \varGamma_i\big)\big] \\ &= \mathrm{E}\Bigg[\mathrm{Q}\Bigg(\frac{-\varPhi_i}{|\varPhi_i|}\sqrt{\frac{\varGamma_i}{g}}(s+\varPhi_i)\Bigg)e^{\frac{\varGamma_i}{g}\left(\frac{s^2}{2}-D_i s\right)} \\ &\quad + \mathrm{Q}\Bigg(\frac{-\varPhi_i}{|\varPhi_i|}\sqrt{\frac{\varGamma_i}{g}}(s-\varPhi_i)\Bigg)e^{\frac{\varGamma_i}{g}\left(\frac{s^2}{2}-(D_i+2\varPhi_i)s\right)}\Bigg], \end{aligned} \tag{2.23}$$

where the outer expectation is taken over D_i, $\varPhi_i$, and $\varGamma_i$. The presence of $|\varPhi_i|$ guarantees that (2.23) is valid for both positive and negative $\varPhi_i$. Note that the random variables D_i and $\varPhi_i$ are highly correlated and should be realized as a pair given by the whole row of Table 2.3 with equal probability. With (2.23), both $\mathrm{M}'(s)$ and $\mathrm{M}''(s)$ can be derived, and the saddlepoint $\hat{s}$ can be solved numerically. In our implementation, we use Newton's method with initial point $s = 1$ because $\hat{s} = 1$ is the saddlepoint[1] for binary-input output-symmetric channels[KL10], e.g. BPSK and QPSK with Gray code.

Note that in (2.22), the Hamming weight $w_{t,l}$ appears both in the exponential term in the numerator as well as the square root in the denominator. This prohibits us from directly using the transfer functions to compute the infinite sum of PEPs in (2.3). Consequently, we approximate the sum $\sum_{t\in\mathbf{T}} \mathrm{P}(\mathbf{e}_{t,l})$ using the

[1] Our saddlepoint for BPSK is $\hat{s} = 1$ as opposed to $\frac{1}{2}$ in [KL10, MFC06] because the moment-generating function is for "half" of log-error-likelihood ratio.

smallest fifteen Hamming weights in the transfer functions [KL10]. The PER is then computed through (2.3).

2.4 Numerical Result

In this section, we compare our PER prediction methods against the PER obtained from an end-to-end Monte Carlo simulation. We will first consider a SISO single carrier system in additive white Gaussian noise (AWGN) and Rayleigh channels, and then look at a 2x2 MIMO OFDM system in frequency selective channels. For illustrative purposes, we have chosen the well-known 64-state rate-1/2 $(133, 171)_8$ convolutional code and the corresponding puncture patterns identified in the IEEE 802.11n standard[Sta09]. However, the analysis is independent of this particular code and can be applied to any other convolutional code and puncture pattern. The packet length L is set to 1024 bytes.

2.4.1 The Comparison Metric

In order to quantify the accuracy of our PER prediction methods and to compare it with the prediction accuracy of other methods, we will use the metric shown below

$$\Delta = \frac{1}{s_2 - s_1} \int_{s_1}^{s_2} \left| \frac{\mathrm{PER}_{\mathrm{pred.}}(s) - \mathrm{PER}_{\mathrm{fit}}(s)}{\mathrm{PER}_{\mathrm{fit}}(s)} \right| \mathrm{d}s, \tag{2.24}$$

where $\mathrm{PER}_{\mathrm{pred.}}(s)$ is the predicted PER as a function of SNR in decibel and the function $\mathrm{PER}_{\mathrm{fit}}(s)$ represents the simulated PER generated by curve-fitting; moreover, s_1 and s_2 are SNR values in decibel satisfying $\mathrm{PER}_{\mathrm{fit}}(s_1) = 0.15$ and $\mathrm{PER}_{\mathrm{fit}}(s_2) = 0.001$. To be more specific, the fitted PER function is given by

$$\mathrm{PER}_{\mathrm{fit}}(s) = \begin{cases} 1 & e^{f(s)} \geq 1 \text{ or } f'(s) > 0 \\ e^{f(s)} & \text{otherwise} \end{cases}, \tag{2.25}$$

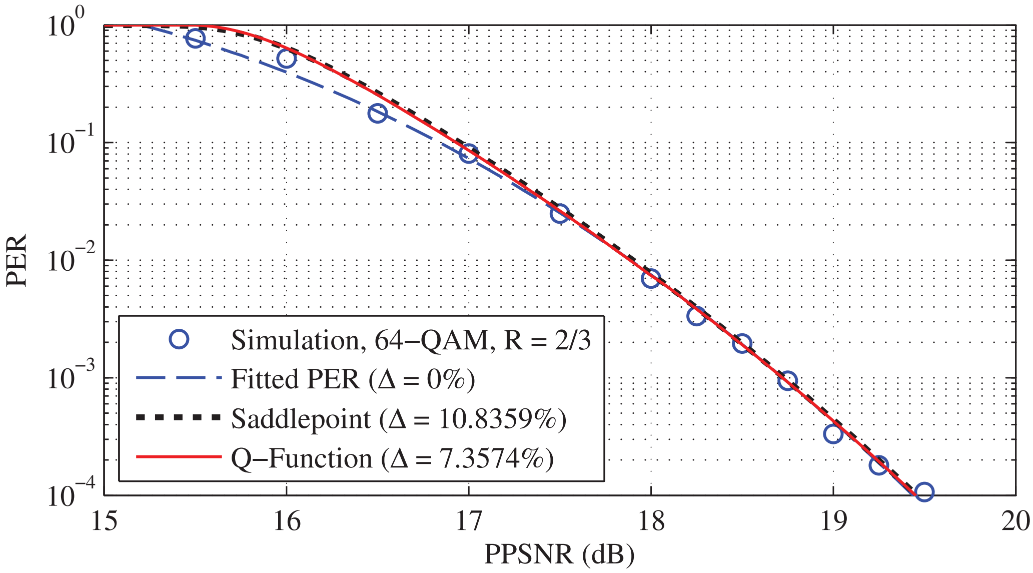


Figure 2.3: Accuracy of analytical PER prediction for 64-QAM rate-2/3 scheme with packet length = 1024 bytes in AWGN channels. The relative prediction error Δ is calculated using (2.24).

where $f(s)$ represents the quadratic fit to the natural logarithm of simulated PER points in the range $0.01\% < \text{PER} < 100\%$ and $f'(s)$ is its first derivative.

Note that in (2.24), the limits of integration are from s_1 to s_2, corresponding to the range of PERs from 0.1% to 15%. This is because in general very few, if any, applications can withstand a PER of 15% or greater. Furthermore, any $\text{PER} < 0.1\%$ is almost as good as 0%. A finer look at (2.24) reveals that it is an averaged relative prediction error for all SNRs in our interest range.

2.4.2 AWGN Channels

The predicted PER of 64-QAM rate-2/3 in an AWGN channel is shown in Fig. 2.3. The relative prediction errors (2.24) of the saddlepoint approximation and Q-function approximation methods are 10.84% and 7.36%, respectively, and mainly due to the prediction errors in the low-SNR region. The reason is that most of

Table 2.5: Relative prediction errors Δ, in percentage (%), of the saddlepoint approximation method (left) and the Q-function approximation method (right) in AWGN channels with packet length = 1024 bytes.

Code Rate	1 / 2	2 / 3	3 / 4	5 / 6
BPSK	8.45, 9.10	10.43, 7.81	8.80, 10.23	7.56, 8.93
QPSK	10.57, 11.20	11.55, 12.26	11.57, 13.01	8.01, 8.38
16-QAM	7.56, 8.25	7.99, 10.10	8.63, 8.36	8.39, 5.46
64-QAM	7.32, 4.33	10.84, 7.36	7.03, 8.72	5.91, 7.57

the assumptions are valid when SNR is high. The relative prediction errors for all combinations of modulations and code rates can be found in Table 2.5. It shows that both analytical methods deliver consistently accurate predictions regardless of the transmission scheme. Note that the heuristic methods are not compared here because for AWGN channels they all use the fitted PER curves as their mapping functions. They will be added to our comparative study when looking at Rayleigh channels and frequency selective channels.

2.4.3 Rayleigh Fading Channels

This section shows the impact of SNR variations brought about by fading. In this set of simulations, we consider a SISO single-carrier system in a quasi-static Rayleigh fading channel. The channel remains constant during a symbol transmission but changes between symbols. Since the bits in a codeword are assumed to be fully interleaved across the fades, the fading coefficients experienced by the symbols can be treated as i.i.d. random variables. In this study, we calculate the PER prediction error (2.24) for the proposed methods as well as the following link quality metrics (LQMs) that appear in the literature: cumulant generating function based effective SNR mapping (κESM)[SGL09], exponential effective SNR mapping (EESM)[Eri03], mutual information effective SNR mapping (MIESM)[TS03], and mean mutual information per coded bit mapping (MMIBM)[JKW10].

Again, the relative prediction error (2.24) is used to quantify the prediction accuracy. However, for the fitted PER function $\mathrm{PER}_{\mathrm{fit}}(s)$ in Rayleigh fading channels, we use a quadratic function for low-to-mid SNR regions and a linear function for high SNR regions as opposed to (2.25). This piecewise function is given by

$$\mathrm{PER}_{\mathrm{fit}}(s) = \begin{cases} 1 & e^{f(s)} \geq 1 \text{ or } f'(s) > 0 \\ e^{f'(\hat{s})(s-\hat{s})+f(\hat{s})} & s > \hat{s} \\ e^{f(s)} & \text{otherwise} \end{cases}, \tag{2.26}$$

where $f(s)$ is still a quadratic function and $\hat{s}$ is the optimized boundary point separating the quadratic fit and the linear fit. The quadratic function $f(s)$ and the boundary point $\hat{s}$ are optimized jointly to minimize the mean square error (MSE) between the natural logarithm of PER points in the range $0.01\% < \mathrm{PER} < 100\%$. Note that, at the boundary point $\hat{s}$, the fitted PER function (2.26) and its first derivative are both continuous.

The relative prediction errors of eight transmission schemes are shown in Fig. 2.4. The heuristic methods might be useful for one scheme but inaccurate for another. Although κESM has the highest averaged prediction error, it is the only LQM that does not require any calibrated parameters. The lowest averaged prediction error among the LQMs is 8.07% while the proposed saddlepoint approximation and Q-function approximation methods have 6.69% and 4.60% averaged prediction errors, respectively.

2.4.4 2x2 Frequency Selective Channels

Finally, we simulate a 2x2 MIMO OFDM system with a zero-forcing (ZF) MIMO decoder operating over a frequency selective channel. The OFDM system is assumed to use a 64-point fast Fourier transform and a subcarrier spacing of

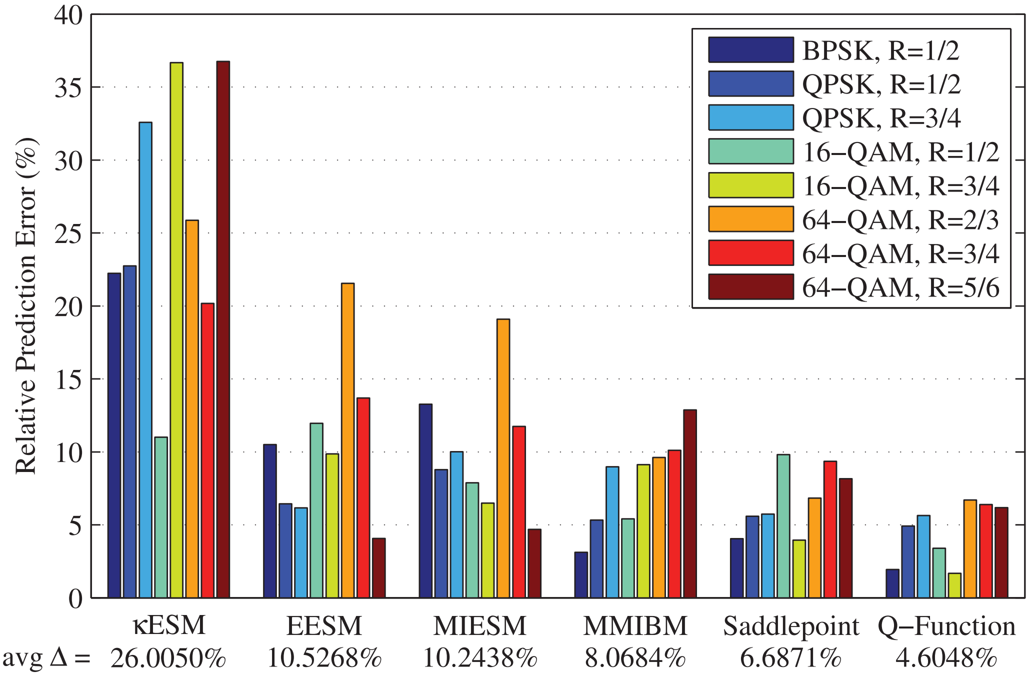


Figure 2.4: Relative PER prediction error Δ for various transmission schemes with packet length $= 1024$ bytes in Rayleigh fading channels. The averaged error is the linear average of the relative prediction errors of all transmission schemes.

312.5kHz. We create the channels based on the TGn channel model A through F[Erc04]. For each channel model, ten realizations are generated independently. For each channel realization, the system is simulated at different SNR levels. Again, we focus on the results in the range $0.1\% < \text{PER} < 15\%$. However, the concept of PER curves does not exist in this scenario because different channel realizations can cause different PERs even under the same averaged SNR. Thus, instead of the relative prediction error (2.24), we calculate the MSE of each method as

$$\text{MSE} = \frac{1}{N}\sum_{n=1}^{N} \left|\ln\left(\text{PER}_{\text{pred.},n}\right) - \ln\left(\text{PER}_{\text{sim.},n}\right)\right|^2, \tag{2.27}$$

where N is the total number of simulation points in the PER range, and $\text{PER}_{\text{pred.},n}$ and $\text{PER}_{\text{sim.},n}$ are the predicted and simulated PER values, respectively, for the n^{th} channel realization.

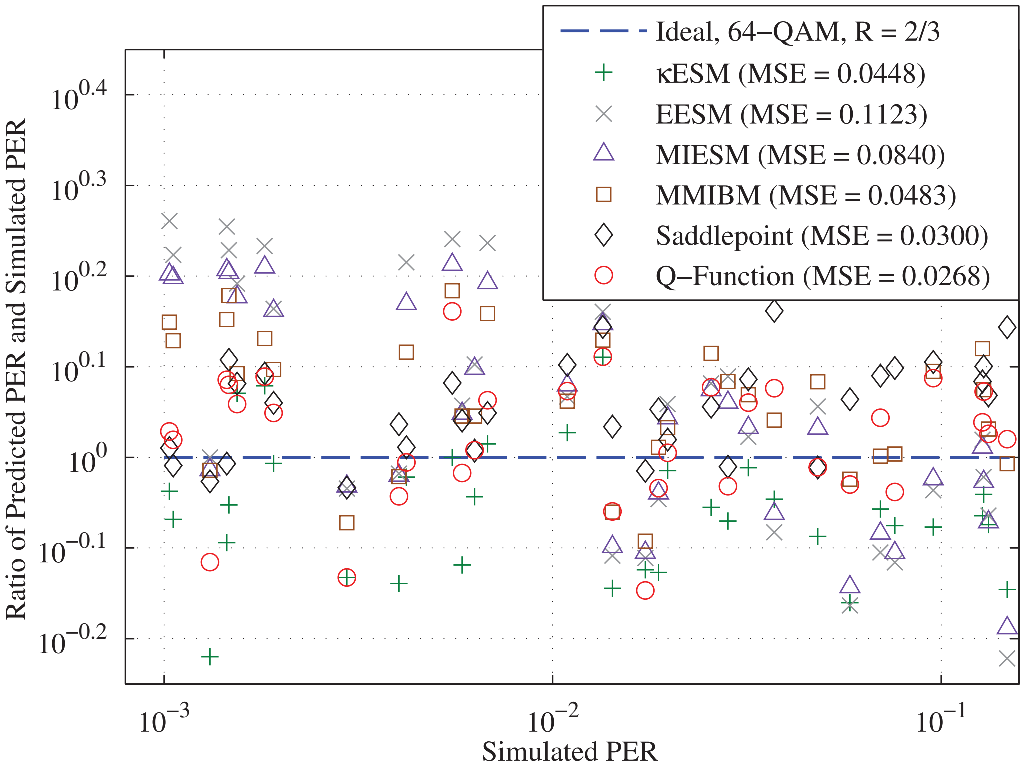


Figure 2.5: Accuracy of PER prediction for 64-QAM rate-2/3 scheme with packet length = 1024 bytes and ZF in 2x2 frequency selective channels generated according to TGn channel model C (τ_{rms} = 30ns). A vertical set of scattered points corresponds to one channel realization predicted by different methods. The mean square error is calculated using (2.27).

In Fig. 2.5, the predicted PER of 64-QAM rate-2/3 scheme operating over a frequency selective channel having an exponential power-delay-profile and an rms-delay spread, $\tau_{\text{rms}} = 30$ns, is illustrated. Only κESM slightly under-estimates the PER and MMIBM over-estimates the PER; EESM and MIESM tend to over-estimate when PER is low and under-estimate when PER is high. The two proposed methods stay around the ideal line with MSEs equal to 0.0300 and 0.0268, respectively. To see the overall accuracy for different transmission schemes under various frequency selectivity, a clear comparison is shown in Fig. 2.6. Each bar corresponds to one MSE value of a prediction method for one transmission scheme

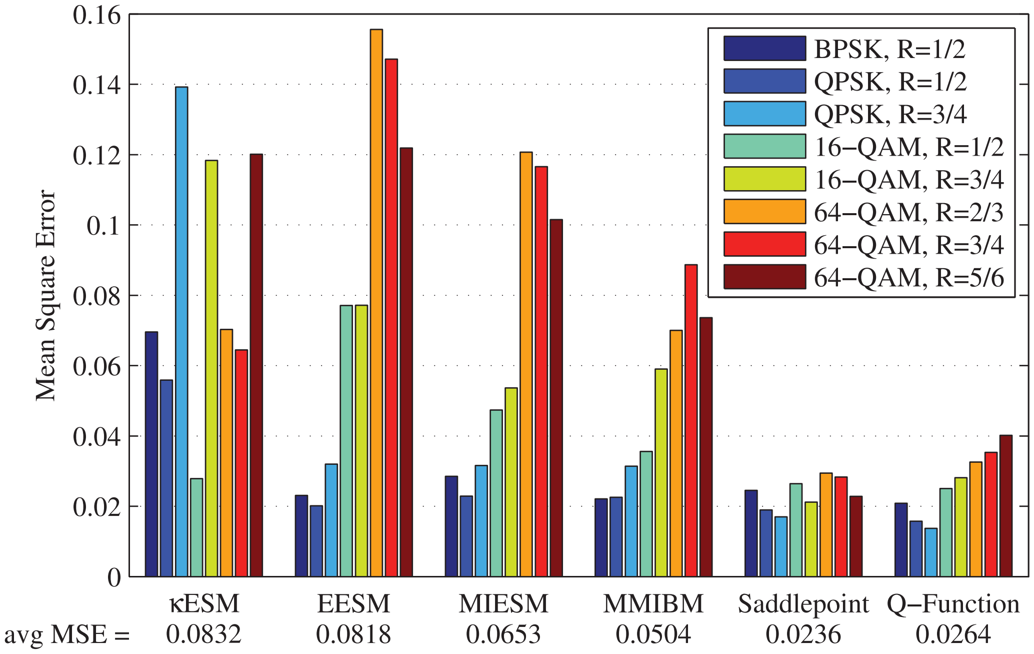


Figure 2.6: Mean square error of PER prediction for various transmission schemes with packet length = 1024 bytes and ZF in 2x2 frequency selective channels having an rms-delay spread τ_{rms} up to 150ns. The averaged MSE is the linear average of the MSEs of all transmission schemes.

in all channel models. The accuracy of LQMs varies a lot from one transmission scheme to another and the lowest averaged MSE among the LQMs is 0.0504 of MMIBM. The proposed saddlepoint approximation and Q-function approximation methods deliver consistently high accuracy for all transmission schemes with averaged MSEs of 0.0236 and 0.0264, respectively.

2.5 Conclusion

In this chapter, we describe two analytical PER prediction methods for punctured convolutional codes based on the codes' transfer functions and the modeling of the distribution of half of log-error-likelihood ratio (HLELR), or essentially LLR. The Q-function approximation method approximates the HLELR using a mixture of normal distributions and relies on the Gaussian Q-function approximation. The saddlepoint approximation method models the HLELR as a mixture of folded normal distributions and applies the saddlepoint approximation. Unlike the existing heuristic methods, the proposed methods require absolutely no curve-fitting or parameter calibrations. Furthermore, they deliver accurate predictions in AWGN, Rayleigh fading, frequency selective fading, and MIMO environments. They are superior to the heuristic methods appearing in the literature.

2.A Approximation of Folded Normal Distribution

In this section, we describe the approximation of a folded normal distribution using a normal distribution satisfying (2.14). For simplicity, denote $\sum_{j=1,j\neq j'}^{d_{\text{free}}} Y^{\star}_{\mathrm{i}(t,l,j)}$ as S. We will first approximate S in (2.14) using a mixture of skew normal distributions [Azz85], and then fine-tune the adjustment function $\mathrm{d}_{\text{adj}}(\cdot)$ of $Y_i^{\star\star}$ in (2.15) by evaluating the targeted conditional probability (2.14).

From (2.13), $Y_i^{\star}$ follows a mixture of folded normal distributions [LNN61]. Furthermore, $Y_i^{\star}$ for all i are i.i.d. because of the uniform random interleaver. Therefore, we are able to evaluate the mean and variance of their sum S based on the conditional PDF (2.9) and the distribution of PPSNR, and then approximate S using a normal distribution. However, by plotting the histogram of S in Fig. 2.7, we observe that the skewness needs to be considered. Thus we try to approximate S using a skew normal distribution [Azz85], which generalizes the normal distribution to allow non-zero skewness. We denote the skew normal distribution as $\mathcal{SN}(\xi, \omega, \lambda)$. Its PDF is

$$f_{\text{SN}}(x) = \frac{2}{\omega} f_{\text{N}}\left(\frac{x-\xi}{\omega}\right) \int\limits_{-\infty}^{\lambda(x-\xi)/\omega} f_{\text{N}}(y)\,\mathrm{d}y, \tag{2.28}$$

where $f_{\text{N}}(x) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{x^2}{-2}\right)$ is the standard normal PDF. In Fig. 2.7, although the skew normal distribution fits much better than the normal distribution, the

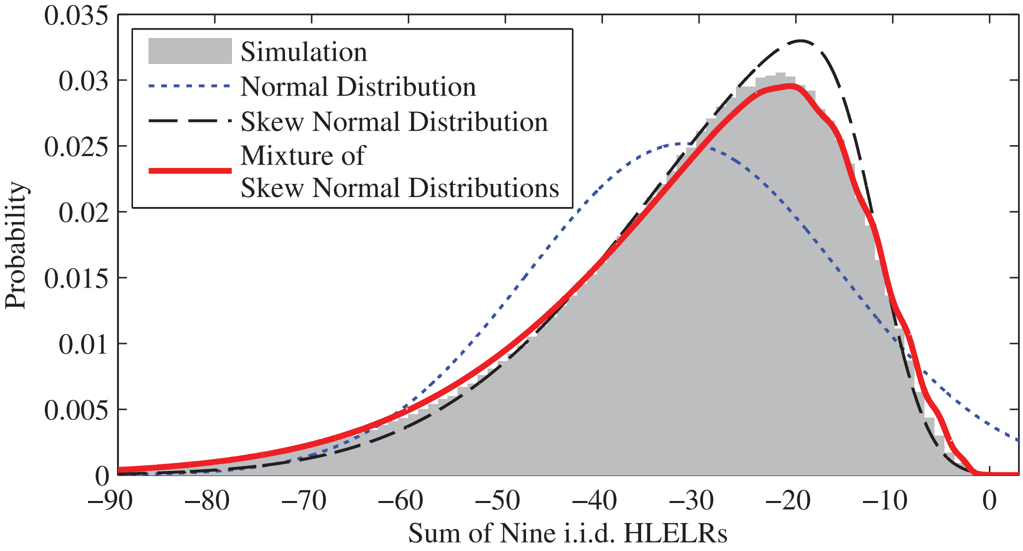


Figure 2.7: The histogram of 10^6 simulated sums of HLELRs. Each sum contains nine i.i.d. HLELRs ($d_{\text{free}} = 10$) of 64-QAM in Rayleigh channel with 15dB averaged PPSNR. The normal distribution matches the mean and variance of the HLELR sum. The skew normal distribution matches the mean, variance, and skewness of the HLELR sum. All PDF curves are properly resized to match the scale of the histogram.

inaccuracy at the tail and the peak is still evident. Thus, similar to approximating $Y_i^\star$ using a mixture of folded normal distributions, we try to approximate S using a mixture of skew normal distributions.

From (2.13), if the realization of $(D_i, \varPhi_i, \varGamma_i)$ is known, the parameters of the associated folded normal distribution are known, and its conditional mean is calculated as

$$\begin{aligned}&\mathrm{E}[Y_i^\star | D_i = d_i, \varPhi_i = \phi_i, \varGamma_i = \gamma_i]\\&= \frac{d_i\gamma_i}{-g} + \frac{\phi_i\gamma_i}{g}\left[\sqrt{\frac{2g}{\pi\phi_i^2\gamma_i}}\exp\left(\frac{\phi_i^2\gamma_i}{-2g}\right) - 2\,\mathrm{Q}\left(\sqrt{\frac{\phi_i^2\gamma_i}{g}}\right)\right].\end{aligned} \tag{2.29}$$

To model the sum of $Y_i^\star$ better, the range of $(D_i, \varPhi_i, \varGamma_i)$ is divided into two clusters

$\mathbf{M}_1$ and $\mathbf{M}_2$, satisfying

$$\mathrm{E}[Y_i^\star | (D_i, \mathit{\Phi}_i, \mathit{\Gamma}_i) \in \mathbf{M}_1] \leq \tilde{\mu} < \mathrm{E}[Y_i^\star | (D_i, \mathit{\Phi}_i, \mathit{\Gamma}_i) \in \mathbf{M}_2]\,,$$

where $\tilde{\mu}$ is the boundary of the two clusters, and the clusters are created by the expectation-maximization algorithm [Bil97]. In general, the conditional distributions of $Y_i^\star$ in a cluster (either $\mathbf{M}_1$ or $\mathbf{M}_2$) tend to be more concentrated than the unconditional distribution of $Y_i^\star$, and hence their sum is closer to a unimodal distribution, which will be modeled as a skew normal distribution.

In S, which is the sum of $d_{\mathrm{free}} - 1$ HLELRs, let random variable H be the number of HLELRs whose parameters lie in $\mathbf{M}_1$, and it is given by

$$H = \sum_{j=1, j \neq j'}^{d_{\mathrm{free}}} \mathrm{I}\big[\big(D_{\mathrm{i}(t,l,j)}, \mathit{\Phi}_{\mathrm{i}(t,l,j)}, \mathit{\Gamma}_{\mathrm{i}(t,l,j)}\big) \in \mathbf{M}_1\big]. \tag{2.30}$$

Since the parameters $(D_i, \mathit{\Phi}_i, \mathit{\Gamma}_i)$ for all i are independent, H follows binomial distribution with parameters $d_{\mathrm{free}} - 1$ and $\mathrm{P}((D_i, \mathit{\Phi}_i, \mathit{\Gamma}_i) \in \mathbf{M}_1)$. The conditional mean, variance, and skewness of S given H can be easily computed from the conditional mean, variance, and skewness of $Y_i^\star$ given $(D_i, \mathit{\Phi}_i, \mathit{\Gamma}_i) \in \mathbf{M}_\theta$, $\theta \in \{1, 2\}$. We further approximate the conditional distribution of S given H using a skew normal distribution $S|(H = h) \sim \mathcal{SN}(\xi_h, \omega_h, \lambda_h)$, where the parameters of the skew normal distribution are chosen such that its mean, variance, and skewness match the conditional mean, variance, and skewness of S given H, respectively. Hence, the unconditioned S is approximated by a mixture of d_{free} skew normal distributions. This mixture distribution fits much better than a single normal or skew normal distribution as illustrated in Fig. 2.7. This approximation is used to evaluate (2.14) and derive the adjustment function $\mathrm{d}_{\mathrm{adj}}(\cdot)$ in (2.15).

The PDF of S is approximated as

$$f_S(x) \approx \sum_{h=0}^{d_{\mathrm{free}}-1} b_h f_{\mathrm{SN},h}(x), \tag{2.31}$$

where $b_h = \mathrm{P}(H = h)$ is the probability mass function (PMF) of the binomial distribution

$$b_h = \binom{d_{\mathrm{free}}-1}{h} \mathrm{P}\big((D_i, \varPhi_i, \varGamma_i) \in \mathbf{M}_1\big)^h \, \mathrm{P}\big((D_i, \varPhi_i, \varGamma_i) \in \mathbf{M}_2\big)^{d_{\mathrm{free}}-1-h}$$

and $f_{\mathrm{SN},h}(x)$ is the PDF of $\mathcal{SN}(\xi_h, \omega_h, \lambda_h)$. Let $\mathbf{U}$ be the collection of random variables $(D_i, \varPhi_i, \varGamma_i)$ and $\mathbf{u}$ be the collection of its realizations (d_i, ϕ_i, γ_i) for the specific $i = \mathrm{i}(t, l, j')$. From (2.13), denote the conditional distribution of $Y_i^\star$ given $\mathbf{U} = \mathbf{u}$ as $\mathcal{FN}(\mu, \rho^2, \eta)$, so its conditional PDF is exactly $f_{\mathrm{FN}}(x)$ given in (2.9), where $\mu = \frac{d_i\gamma_i}{-g}$, $\rho = \sqrt{\frac{\gamma_i}{g}}$, and $\eta = \frac{\phi_i\gamma_i}{g}$. Thus the left-hand side (LHS) of (2.14) is given by

$$\mathrm{P}(Y_i^\star + S > 0 | \mathbf{U} = \mathbf{u}) = \int_{-\infty}^{\infty} f_{\mathrm{FN}}(x) \int_{-x}^{\infty} f_S(y)\,\mathrm{d}y\,\mathrm{d}x. \tag{2.32}$$

By evaluating the folded normal PDF, (2.32) leads to

$$\begin{aligned}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\rho^2}} e^{\frac{(x-\mu)^2}{-2\rho^2}} \int_{-x}^{\infty} f_S(y)\,\mathrm{d}y\,\mathrm{d}x + \frac{|\eta|}{\eta} \int_{\mu-\eta}^{\infty} \frac{1}{\sqrt{2\pi\rho^2}} e^{\frac{(x-\mu+\eta+|\eta|)^2}{-2\rho^2}} \int_{-x}^{\infty} f_S(y)\,\mathrm{d}y\,\mathrm{d}x \\ - \frac{|\eta|}{\eta} \int_{-\infty}^{\mu-\eta} \frac{1}{\sqrt{2\pi\rho^2}} e^{\frac{(x-\mu+\eta-|\eta|)^2}{-2\rho^2}} \int_{-x}^{\infty} f_S(y)\,\mathrm{d}y\,\mathrm{d}x,\end{aligned} \tag{2.33}$$

where the use of $|\eta|$ makes sure the validity for both positive and negative η. Using changes of variables, (2.33) is simplified to

$$\begin{aligned}\int_{-\infty}^{\infty} f_{\mathrm{N}}(x') \int_{-\rho x'-\mu}^{\infty} f_S(y)\,\mathrm{d}y\,\mathrm{d}x' + \frac{|\eta|}{\eta} \int_{-\infty}^{-|\eta|/\rho} f_{\mathrm{N}}(x'') \int_{\rho x''-\mu+\eta+|\eta|}^{\infty} f_S(y)\,\mathrm{d}y\,\mathrm{d}x'' \\ - \frac{|\eta|}{\eta} \int_{-\infty}^{-|\eta|/\rho} f_{\mathrm{N}}(x''') \int_{-\rho x'''-\mu+\eta-|\eta|}^{\infty} f_S(y)\,\mathrm{d}y\,\mathrm{d}x'''.\end{aligned} \tag{2.34}$$

Moreover, with (2.31), the second and the third terms of (2.34) are given by the weighted sum of

$$\int_{-\infty}^{-|\eta|/\rho} f_{\mathrm{N}}(x) \int_{\pm\rho x-\mu+\eta\pm|\eta|}^{\infty} f_{\mathrm{SN},h}(y)\,\mathrm{d}y\,\mathrm{d}x$$

$$= 2 \int_{-\infty}^{-|\eta|/\rho} f_{\mathrm{N}}(x) \int_{(\pm\rho x - \mu + \eta \pm |\eta| - \xi_h)/\omega_h}^{\infty} f_{\mathrm{N}}(y') \int_{-\infty}^{\lambda_h y'} f_{\mathrm{N}}(z)\, \mathrm{d}z\, \mathrm{d}y' \mathrm{d}x, \tag{2.35}$$

where the signs of ρ and $|\eta|$ depend on whether it represents the second or the third term of (2.34) and another change of variables is used in (2.35). After further changes of variables $y'' = -y' \pm \frac{\rho}{\omega_h} x$ and $z' = z - \lambda_h y'$, (2.35) yields twice as much as the cube volume $\left\{ x < \frac{-|\eta|}{\rho}, y'' < \frac{\mu - \eta \mp |\eta| + \xi_h}{\omega_h}, z' < 0 \right\}$ of a trivariate normal distribution with zero mean and covariance

$$\begin{bmatrix} 1 & \pm\rho/\omega_h & 0 \\ \pm\rho/\omega_h & 1 + (\rho/\omega_h)^2 & \lambda_h \\ 0 & \lambda_h & 1 + \lambda_h^2 \end{bmatrix}.$$

Since the method to evaluate the cumulative distribution function of a multivariate normal distribution is well studied [GB09], the second and the third terms of (2.34) can be calculated.

In the right-hand side (RHS) of (2.14), define

$$S_i' \triangleq Y_i^{\star\star} + S + \frac{\varGamma_i}{g} \mathrm{d}_{\mathrm{adj}} \left(\frac{D_i \gamma_i}{-g}, \frac{\varGamma_i}{g}, \frac{\varPhi_i \varGamma_i}{g} \right).$$

Similar to the conditional distribution of S given H, the conditional distribution of S_i' given $H = h$ and $\mathbf{U} = \mathbf{u}$ can be approximated by a skew normal distribution $\mathcal{SN}(\xi_h', \omega_h', \lambda_h')$. The parameters ξ_h', ω_h', and λ_h' are chosen such that the mean, variance, and skewness of the skew normal distribution match the conditional mean, variance, and skewness of S_i', respectively. Thus the conditional PDF of S_i' given $\mathbf{U} = \mathbf{u}$ is

$$f_{S_i'}(x) \approx \sum_{h=0}^{d_{\mathrm{free}}-1} b_h f_{\mathrm{SN},h,i}(x),$$

where $f_{\mathrm{SN},h,i}(x)$ is the PDF of $\mathcal{SN}(\xi_h', \omega_h', \lambda_h')$. Denote $\frac{\varGamma_i}{g} \mathrm{d}_{\mathrm{adj}} \left(\frac{D_i \gamma_i}{-g}, \frac{\varGamma_i}{g}, \frac{\varPhi_i \varGamma_i}{g} \right)$ as ζ, and then the RHS of (2.14) can be written as $\mathrm{P}(S_i' > \zeta | \mathbf{U} = \mathbf{u})$. Since the adjustment ζ is close to zero, this tail probability can be approximated by its second-order Taylor series around zero as

$$\int_{\zeta}^{\infty} f_{S_i'}(x)\mathrm{d}x \approx \int_{0}^{\infty} f_{S_i'}(x)\mathrm{d}x - f_{S_i'}(0)\zeta - \frac{\frac{\mathrm{d}}{\mathrm{d}x} f_{S_i'}(x)\big|_{x=0}}{2}\zeta^2. \tag{2.36}$$

The zeroth order term of (2.36) approximates the probability

$$\mathrm{P}(Y_i^{\star\star} + \zeta + S > 0 | \mathbf{U} = \mathbf{u})\,, \tag{2.37}$$

where the conditional distribution of $Y_i^{\star\star}+\zeta$ given $\mathbf{U} = \mathbf{u}$ is actually $\mathcal{N}\Big(\frac{d_i\gamma_i}{-g}, \frac{\gamma_i}{g}\Big) = \mathcal{N}(\mu, \rho^2)$. Therefore, the probability in (2.37) is exactly given by the first term of (2.34). After combining the LHS (2.34) and RHS (2.36) of (2.14), we obtain the equation

$$\frac{\frac{\mathrm{d}}{\mathrm{d}x} f_{S_i'}(x)\big|_{x=0}}{2}\zeta^2 + f_{S_i'}(0)\zeta + J \approx 0, \tag{2.38}$$

where J denotes the sum of the second and the third terms of (2.34). The correct ζ should be the root which is closer to zero, and then the value of the adjustment function $\mathrm{d}_{\mathrm{adj}}(\mu, \rho^2, \eta)$ is known. Note that the the adjustment function should has opposite sign to L and η. In case the real root does not exist due to rounding errors, use the first-order Taylor series and it leads to $\zeta = -J/f_{S_i'}(0)$.

We provide an alternative approximation of the adjustment function

$$\mathrm{d}_{\mathrm{adj}}\big(\mu, \rho^2, \eta\big) \approx \frac{2\eta}{\rho^2}\,\mathrm{Q}\Big(\frac{|\eta|}{\rho}\Big)\Bigg[2\,\mathrm{Q}\Big(\frac{|\eta|}{\rho}\Big) - \sqrt{\frac{2\rho^2}{\pi\eta^2}}\exp\Big(\frac{-\eta^2}{2\rho^2}\Big)\Bigg]. \tag{2.39}$$

Compared to obtaining and solving (2.38), this approximation is much easier to compute, but might not be applicable to all convolutional codes (d_{free}).

2.B Transfer Functions of Punctured Convolutional Codes

In this section, we present the computation of the transfer functions of punctured convolutional codes. Although there has been studies on this topic, we have not yet seen any that allows multiple transfer functions for one code. In the past, the transfer function included all error events starting at all puncture states, i.e.

$\sum_{l=1}^{M} \mathrm{T}_l(W)$, which was only useful to analyze the bit error rate (BER) but not the PER. Even if one uses its average $\frac{1}{M}\sum_{l=1}^{M} \mathrm{T}_l(W)$ for the PER analysis, this is still an approximation because the coefficients may not be whole numbers. We will propose two methods to compute the exact transfer functions. The first method is straightforward, while the second method is preferred when the number of puncture states is large. Both proposed methods are based on the same technique, which is used to find the transfer function of a non-punctured convolutional code [McE98], but applied to different diagrams.

2.B.1 Extended State Method

When puncturing is applied, we may extend the state space to incorporate the puncture states [BHP90]. An example of the extended error-state diagram of the $(7,5)_8$ convolutional code punctured by the pattern $\left[\begin{smallmatrix}1&1\\1&0\end{smallmatrix}\right]$ is shown in Fig. 2.8b, where the state is denoted by (puncture state, encoder state). The states S_0 and S_4 represent the start states of error events beginning at the first and second puncture states, respectively. Therefore, $\mathrm{T}_1(W)$ is obtained by summing all paths from S_0 to S_0' or S_4'. The transition among the intermediate states can be written as

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_5 \\ S_6 \\ S_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & W & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & W \\ 0 & 1 & 0 & 0 & 0 & 0 \\ W & 0 & W & 0 & 0 & 0 \\ W & 0 & W & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_5 \\ S_6 \\ S_7 \end{bmatrix}.$$

Let $\bar{A}^{\mathrm{ext}}$ be the transition matrix, $\bar{A}^{\mathrm{ext}}_{\mathrm{end}} = [0, W^2, 0, 0, W, 0]$ be the transition to the end states, $\bar{A}^{\mathrm{ext}}_{\mathrm{begin},1} = [0,0,0,W^2,0,0]^{\mathrm{T}}$ be the transition from S_0, and $\bar{A}^{\mathrm{ext}}_{\mathrm{begin},2} = [W,0,0,0,0,0]^{\mathrm{T}}$ be the transition from S_4, where T represents the matrix transpose and the superscript "ext" labels the matrices created from the extended error-state

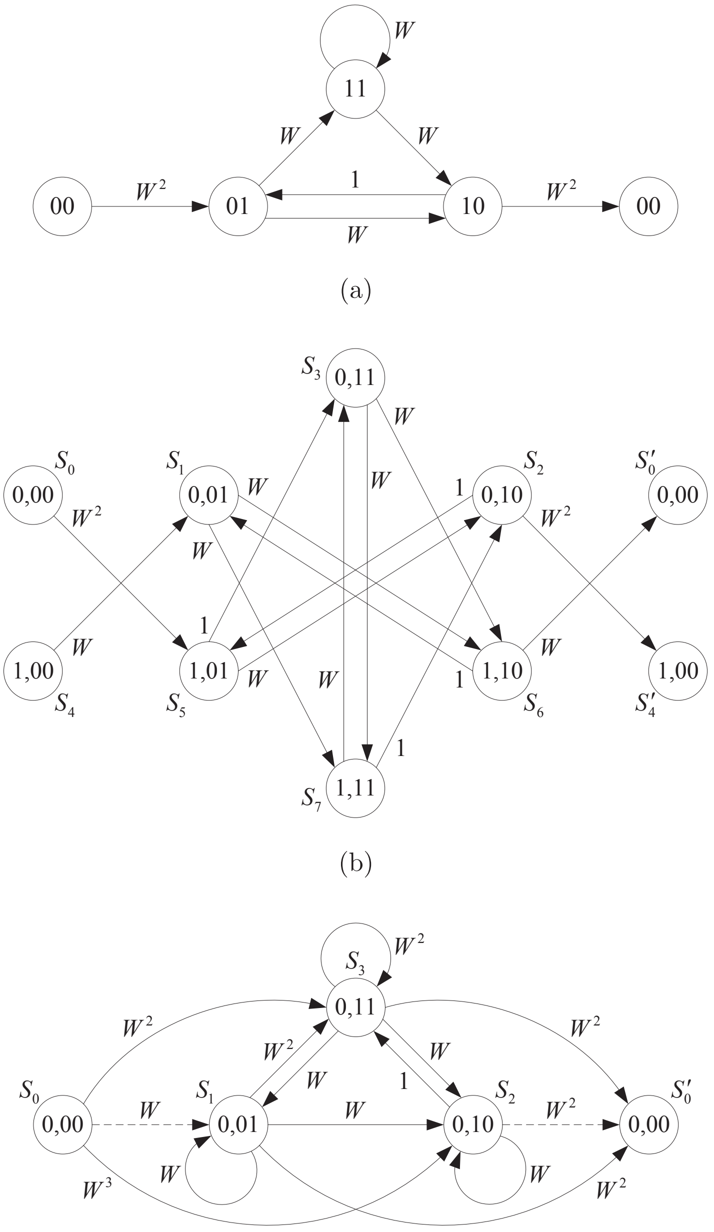


Figure 2.8: The error-state diagram (a) of the $(7,5)_8$ convolutional code. Its extended error-state diagram (b) and condensed error-state diagram (c) when puncture pattern $[1,1;1,0]$ is applied. The exponent of W records the output Hamming weight of each transition.

diagram. Hence, the transfer functions are given by

$$\mathrm{T}_1(W) = \bar{A}^{\text{ext}}_{\text{end}} \left(\bar{I}_6 - \bar{A}\right)^{-1} \bar{A}^{\text{ext}}_{\text{begin},1}$$

and

$$\mathrm{T}_2(W) = \bar{A}_{\text{end}}^{\text{ext}} \left(\bar{I}_6 - \bar{A}\right)^{-1} \bar{A}_{\text{begin},2}^{\text{ext}},$$

where $\bar{I}_k$ is the $k \times k$ identity matrix.

In general, consider an N-state convolutional code punctured by an M-state puncturer. There are $M\,(N-1)$ intermediate states and the transfer functions are given by

$$\mathrm{T}_l(W) = \bar{A}_{\text{end}}^{\text{ext}} \left(\bar{I}_{M(N-1)} - \bar{A}^{\text{ext}}\right)^{-1} \bar{A}_{\text{begin},l}^{\text{ext}},\ 1 \le l \le M, \tag{2.40}$$

where all matrices stem from its extended error-state diagram. However, this requires a size $M\,(N-1) \times M\,(N-1)$ matrix inversion, which could be computationally intensive. Fortunately, this can be simplified because of the sparse structure of $\bar{A}^{\text{ext}}$,

$$\bar{A}^{\text{ext}} = \begin{bmatrix} \bar{0} & \bar{0} & & \cdots & \bar{0} & \bar{A}_M \\ \bar{A}_1 & \bar{0} & \bar{0} & & \vdots & \bar{0} \\ \bar{0} & \bar{A}_2 & \bar{0} & \ddots & & \bar{0} \\ & \bar{0} & \bar{A}_3 & \ddots & \bar{0} & \vdots \\ \vdots & & \ddots & \ddots & \bar{0} & \bar{0} \\ \bar{0} & \cdots & & \bar{0} & \bar{A}_{M-1} & \bar{0} \end{bmatrix}, \tag{2.41}$$

where each submatrix is $(N-1) \times (N-1)$, $\bar{0}$ is a zero matrix, and $\bar{A}_l$ corresponds to the transition from puncture state $l-1$ to puncture state l modulo M for all l. Let

$$\left(\bar{I}_{M(N-1)} - \bar{A}^{\text{ext}}\right)^{-1} = \begin{bmatrix} \bar{B}_{1,1} & \bar{B}_{1,2} & \cdots & \bar{B}_{1,M} \\ \bar{B}_{2,1} & \bar{B}_{2,2} & \cdots & \bar{B}_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{B}_{M,1} & \bar{B}_{M,2} & \cdots & \bar{B}_{M,M} \end{bmatrix},$$

where all $\bar{B}_{i,j}$ are $(N-1) \times (N-1)$ matrices. Using Gaussian elimination, we have $\bar{B}_{i,j} = \bar{C}_{i,j} + \bar{D}_{i,j}$, where

$$\bar{C}_{i,j} = \begin{cases} 0 & i < j \\ \bar{I}_{N-1} & i = j \\ \bar{A}_{i-1}\bar{A}_{i-2} \cdots \bar{A}_j & i > j \end{cases} \tag{2.42}$$

and

$$\bar{D}_{i,j} = \bar{A}_{i-1}\bar{A}_{i-2} \cdots \bar{A}_0 \bar{E} \bar{A}_M \bar{A}_{M-1} \cdots \bar{A}_j. \tag{2.43}$$

In (2.43), we define $\bar{E} \triangleq \left(\bar{I}_{N-1} - \bar{A}_M \bar{A}_{M-1} \cdots \bar{A}_1\right)^{-1}$ and $\bar{A}_0 \triangleq \bar{I}_{N-1}$. Thus it only requires an $(N-1) \times (N-1)$ matrix inversion, which has the same size as the transfer function calculation of a non-punctured convolutional code and is independent of the number of puncture states. Furthermore, $C_{i,j}$ and $D_{i,j}$ for all i and j have lots of common terms and can be reused to ease the computation. For example, we can begin with $\bar{B}_{1,M} = \bar{E}\bar{A}_M$, and calculate the rest recursively by following

$$\bar{B}_{i+1,j} = \bar{A}_i \bar{B}_{i,j} + \bar{I}_{N-1}\,\mathrm{I}(i+1 = j) \tag{2.44a}$$

$$\bar{B}_{i,j-1} = \bar{B}_{i,j} \bar{A}_{j-1} + \bar{I}_{N-1}\,\mathrm{I}(i = j-1)\,, \tag{2.44b}$$

where again $\mathrm{I}(\cdot)$ is the indicator function.

Though the recursive algorithm simplifies the computation, it still requires about M^2 matrix multiplications of size $(N-1) \times (N-1)$. Thus it motivates us to reduce the number of states.

2.B.2 Condensed State Method

To reduce the number of states, we may combine M state transitions as a big transition. In this way, if a big transition initiates from puncture state zero, it will still lead to puncture state zero. Therefore, the number of states does not increase as puncturing is applied. This idea was found in [HB89], but it failed to deliver the exact transfer function. The reason is that even though both the start and end states of a big transition are not the all-zero state, the path might have visited encoder state zero, which terminates error events, during the big transition. As a result, such error events get counted multiple times, and this explains why the weight spectra in [HB89] is greater or equal to which in [BHP90]. The same mistake was made in [LK04] even if the exact state diagram of a punctured convolutional code was proposed.

An example of the condensed error-state diagram of the same punctured convolutional code as in the previous subsection is shown in Fig. 2.8c. Each line corresponds to $M = 2$ state transitions, and the dashed lines include one transition inside encoder state zero. Note, although the big transition $S_2 \to S_1$ is possible by having [01] as the input, the path will first visit S'_4 in Fig. 2.8b, which terminates error events, and should not leave encoder state zero in the error-state diagram. Thus, this big transition should be avoided because these error events are incorporated in the transition $S_2 \to S'_0$. Since the difference between transfer functions lies in the first transition, we will compute $\mathrm{T}_1(W)$ by summing all paths beginning with transitions $S_0 \to S_2$ and $S_0 \to S_3$, and derive $\mathrm{T}_2(W)$ by starting with $S_0 \to S_1$. The transition among the intermediate states is

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} W & 0 & W \\ W & W & W \\ W^2 & 1 & W^2 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix}.$$

Let $\bar{A}^{\mathrm{cond}}$ be the transition matrix, $\bar{A}^{\mathrm{cond}}_{\mathrm{end}} = [W^2, W^2, W^2]$ be the transition to the end state, $\bar{A}^{\mathrm{cond}}_{\mathrm{begin},1} = [0, W^3, W^2]^{\mathrm{T}}$ be the first transition of $\mathrm{T}_1(W)$, and $\bar{A}^{\mathrm{cond}}_{\mathrm{begin},2} = [W, 0, 0]^{\mathrm{T}}$ be the first transition of $\mathrm{T}_2(W)$. We have transfer functions

$$\begin{aligned} \mathrm{T}_1(W) &= \bar{A}^{\mathrm{cond}}_{\mathrm{end}} \left(\bar{I}_3 - \bar{A}^{\mathrm{cond}}\right)^{-1} \bar{A}^{\mathrm{cond}}_{\mathrm{begin},1} \\ \mathrm{T}_2(W) &= \bar{A}^{\mathrm{cond}}_{\mathrm{end}} \left(\bar{I}_3 - \bar{A}^{\mathrm{cond}}\right)^{-1} \bar{A}^{\mathrm{cond}}_{\mathrm{begin},2}. \end{aligned}$$

In general, for an N-state convolutional code punctured by an M-state puncturer, the transfer functions are given by

$$\mathrm{T}_l(W) = \bar{A}_{\text{end}}^{\text{cond}} \left(\bar{I}_{N-1} - \bar{A}^{\text{cond}}\right)^{-1} \bar{A}_{\text{begin},l}^{\text{cond}}, \;\; 1 \leq l \leq M, \tag{2.45}$$

where $\bar{A}$, $\bar{A}_{\text{end}}$, and $\bar{A}_{\text{begin},l}$ for all l are specified according to its condensed error-state diagram. Note that the relationship between transition matrices $\bar{A}^{\text{ext}}$ and $\bar{A}^{\text{cond}}$ is $\bar{A}^{\text{cond}} = \bar{A}_M \bar{A}_{M-1} \cdots \bar{A}_1$, where $\bar{A}_l$ for all l are given in (2.41). Since (2.45) only needs one $(N-1) \times (N-1)$ matrix inversion, it is preferable to (2.40) especially for a large M.

CHAPTER 3

Analytical Packet Error Rate Predictions of Punctured Convolutionally Coded Systems with a Repeated-Pattern Interleaver

In the previous chapter, we have presented the analytical packet error rate (PER) prediction methods for punctured convolutionally coded systems with a uniform random interleaver, where the bits' log-likelihood ratios (LLRs) are assumed to be independent and identically distributed (i.i.d.) random variables. In a practical system, the interleaver usually has a fixed permutation pattern and thus the assumption is not valid anymore. We consider a multiple-input multiple-output (MIMO) orthogonal frequency-division multiplexing (OFDM) system employing a repeated-pattern interleaver, and discuss three types of LLRs' correlation. The three types include a) correlation across subcarriers, b) correlation across spatial streams, and c) correlation within a subcarrier. The simulation result shows that our analytical method delivers significantly higher accuracy than all the other existing PER prediction methods. Unlike the heuristic methods, the proposed analytical method requires no curve-fitting or parameter calibrations.

This chapter is organized as follows: Section 3.1 presents the model of a MIMO OFDM system employing a repeated-pattern interleaver. Section 3.2 recapitulates the analytical PER prediction method for systems employing a uniform random interleaver. Section 3.3 generalizes the analytical PER prediction to the usage of a repeated-pattern interleaver by considering three types of correlation of LLRs. In

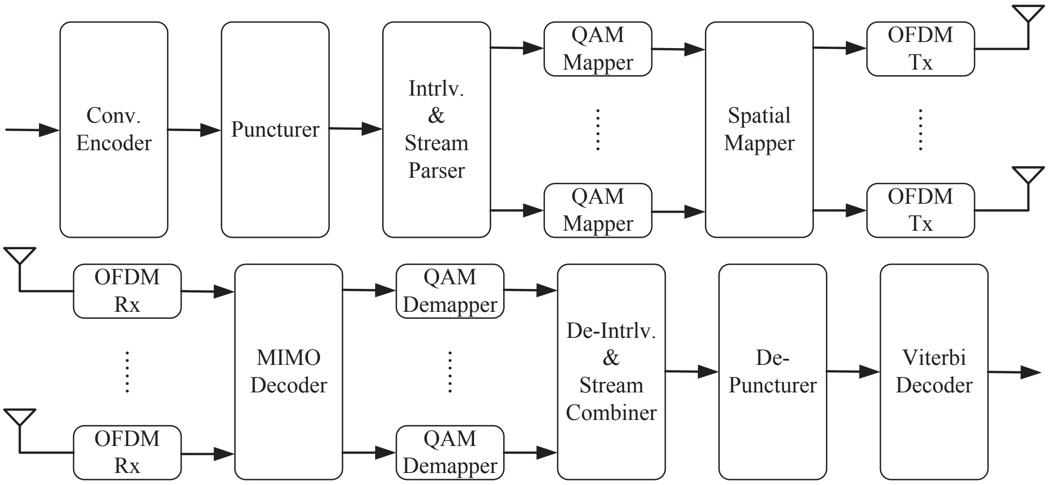


Figure 3.1: The block diagram of a MIMO OFDM system employing punctured convolutional codes with BICM.

Section 3.4, the accuracy of the proposed method is compared with other existing methods. We conclude this chapter in Section 3.5.

Notation: Boldface letters $\mathbf{x}$ represent sets. Vectors and matrices are denoted by letters with a bar $\bar{x}$. Random variables are written as italic uppercase letters X, and their realizations are represented by the corresponding italic lowercase letters x.

3.1 System Model

Fig. 3.1 illustrates a MIMO OFDM system employing a punctured convolutional code with bit-interleaved coded modulation (BICM). In this system, a length-L information bit sequence, constituting a packet, is fed into a convolutional encoder, and the coded bit sequence can be punctured to reach a higher code rate. The punctured bit sequence $\bar{X}$ is then interleaved by a repeated-pattern interleaver and spread over one or more spatial streams. The repeated-pattern interleaver uses a fixed permutation pattern for each OFDM symbol in a packet, and this pattern is fixed for all packet transmissions. Note that, depending on the transmission scheme (or the number of bits in an OFDM symbol), the permutation pattern could be different.

Afterward the bit sequence of each spatial stream is modulated using binary phase-shift keying (BPSK), quadrature phase-shift keying (QPSK), 16-quadrature amplitude modulation (QAM), or 64-QAM with Gray code, and the mapping rule follows IEEE 802.11n standard [Sta09].

We assume that the channel coefficients and the noise spectral density N_0 are known. Therefore, the signal-to-noise ratio (SNR) of each modulation symbol at the receiver input is known. We put no constraint on the antenna configurations; the system can use either single antenna or multiple antennae with MIMO techniques, e.g., beamforming, spatial multiplexing, and/or space-time code. However, we do assume that the effective noise after the MIMO decoder is independent of the transmitted information, i.e., no inter-symbol interference, inter-carrier interference, or interference between spatial streams. With the knowledge of the channel coefficients, noise spectral density, and the MIMO technique being used, the post-processing signal-to-noise ratio (PPSNR)[HSP01], which is defined as the SNR right before the symbol demapper, of each modulation symbol can be calculated. Note that the noise before the demapper, belonging to the same subcarrier and OFDM symbol but different spatial streams, could be dependent due to the MIMO decoder. The receiver is equipped with a simplified soft-demapper[TB02], which is an approximation of the max-log demapper[Vit98] but requires rather simple computations. Finally, the deinterleaved LLR sequence $\bar{Y}$ is depunctured and decoded by a soft decision Viterbi decoder.

In the rest of the chapter, we will use the system following IEEE 802.11n standard[Sta09] as an example, including the specified convolutional code, puncture pattern, interleaver, modulation, and OFDM parameters. Regarding the MIMO technique, we use spatial multiplexing with N_s spatial streams and N_s-by-N_s antenna configurations. The receiver is equipped with a zero-forcing (ZF)

MIMO decoder to prevent any interference between spatial streams. Note that, in spite of the setting of the example, the analysis is applicable to all punctured convolutionally coded MIMO OFDM systems with a repeated-pattern interleaver.

3.2 Analytical PER Prediction for Systems with a Uniform Random Interleaver

In this section, we summarize the results of the analytical PER prediction for systems with a uniform random interleaver presented in the previous chapter. They serve as the fundamental building blocks of the analytical PER prediction for systems with a repeated-pattern interleaver.

The PER of a length-L packet is approximated by

$$\mathrm{PER}_L \approx 1 - \prod_{l=1}^{L} \left[1 - \sum_{t \in \mathbf{T}} \mathrm{P}(\mathbf{e}_{t,l}) \right], \tag{3.1}$$

where $\mathbf{T}$ is the set of indices of all error events starting at the same location without limiting the length of the error events, and $\mathbf{e}_{t,l}$ is the pairwise error event starting at the l^{th} transition and indexed as t. We define half of log-error-liklihodd ratio (HLELR) as

$$Y_i^\star \triangleq Y_i \, \mathrm{sgn}(X_i) \, / \, 2 \; , \tag{3.2}$$

where Y_i and X_i are the i^{th} elements of the deinterleaved LLR sequence $\bar{Y}$ at the receiver and the punctured bit sequence $\bar{X}$ at the transmitter, respectively, and $\mathrm{sgn}(\cdot)$ is a function which maps 1 to 1 and 0 to -1. The probability of a pairwise error event, called the pairwise error probability (PEP), is then given by

$$\mathrm{P}(\mathbf{e}_{t,l}) = \mathrm{P}\left(\sum_{j=1}^{w_{t,l}} Y^\star_{\mathrm{i}(t,l,j)} > 0 \right) , \tag{3.3}$$

where $w_{t,l}$ is the Hamming weight of this error event, and the function $\mathrm{i}(t,l,j)$ outputs the location of the j^{th} nonzero bit of the sequence associated with the pairwise error event $\mathbf{e}_{t,l}$.

Table 3.1: The possible realization pairs of $(D_{\mathrm{I},k}, \mathit{\Phi}_{\mathrm{I},k})$ and the modulation dependent scaling factor g.

Modulation	$(d_{\mathrm{I},1}, \phi_{\mathrm{I},1})$	$(d_{\mathrm{I},2}, \phi_{\mathrm{I},2})$	$(d_{\mathrm{I},3}, \phi_{\mathrm{I},3})$	g
BPSK	$(1, \infty)$	N/A	N/A	0.5
QPSK	$(1, \infty)$	N/A	N/A	1
16-QAM	$(1, \infty), (3, \infty)$	$(1, 1), (1, -3)$	N/A	5
64-QAM	$(1, \infty), (3, \infty),$ $(5, \infty), (7, \infty)$	$(3, 1), (1, 3),$ $(1, -5), (3, -7)$	$(1, 1), (1, 1),$ $(1, 1), (1, -3)$	21

We showed that the conditional distribution of an HLELR given its parameter pair $(D_i, \mathit{\Phi}_i)$ and its PPSNR $\mathit{\Gamma}_i$ follows a folded normal distribution

$$Y_i^\star | (D_i = d_i, \mathit{\Phi}_i = \phi_i, \mathit{\Gamma}_i = \gamma_i) \sim \mathcal{FN}\left(\frac{d_i\gamma_i}{-g}, \frac{\gamma_i}{g}, \frac{\phi_i\gamma_i}{g}\right), \tag{3.4}$$

where g is a scaling factor depending on the modulation scheme and given in Table 3.1. The parameter pair $(D_i, \mathit{\Phi}_i)$ is a pair of random variables and are a mixture of random variable pairs $(D_{\mathrm{I},k}, \mathit{\Phi}_{\mathrm{I},k})$ in Table 3.1 for all k with equal weights.

We further approximate the HLELR $Y_i^\star$ using a random variable $Y_i^{\star\star}$. The conditional distribution of $Y_i^{\star\star}$ given the parameter pair $(D_i, \mathit{\Phi}_i)$ and its PPSNR $\mathit{\Gamma}_i$ follows a normal distribution

$$Y_i^{\star\star} | (D_i = d_i, \mathit{\Phi}_i = \phi_i, \mathit{\Gamma}_i = \gamma_i) \sim \mathcal{N}\left(-\left[d_i + \mathrm{d_{adj}}\left(\frac{d_i\gamma_i}{-g}, \frac{\gamma_i}{g}, \frac{\phi_i\gamma_i}{g}\right)\right]\frac{\gamma_i}{g}, \frac{\gamma_i}{g}\right), \tag{3.5}$$

where the adjustment function $\mathrm{d_{adj}}(\cdot)$ is given in Appendix 2.A. For simplicity, we further define random variables

$$D'_j \triangleq D_{\mathrm{i}(t,l,j)} + \mathrm{d_{adj}}\left(\frac{D_{\mathrm{i}(t,l,j)}\mathit{\Gamma}_{\mathrm{i}(t,l,j)}}{-g}, \frac{\mathit{\Gamma}_{\mathrm{i}(t,l,j)}}{g}, \frac{\mathit{\Phi}_{\mathrm{i}(t,l,j)}\mathit{\Gamma}_{\mathrm{i}(t,l,j)}}{g}\right)$$

and $\mathit{\Gamma}'_j \triangleq \mathit{\Gamma}_{\mathrm{i}(t,l,j)}/g$. Through the Gaussian Q-function approximation[CDS03], the PEP is then approximated as

$$\mathrm{P}(\mathbf{e}_{t,l}) \approx \sum_{r=1}^{4} \alpha_r \, \mathrm{E}\Big[e^{-\beta_r \mathit{\Gamma}'\{D' + (D'-1)\,\mathrm{E}[D']\}}\Big]^{w_{t,l}}, \tag{3.6}$$

Table 3.2: The coefficients of Gaussian Q-function approximation.

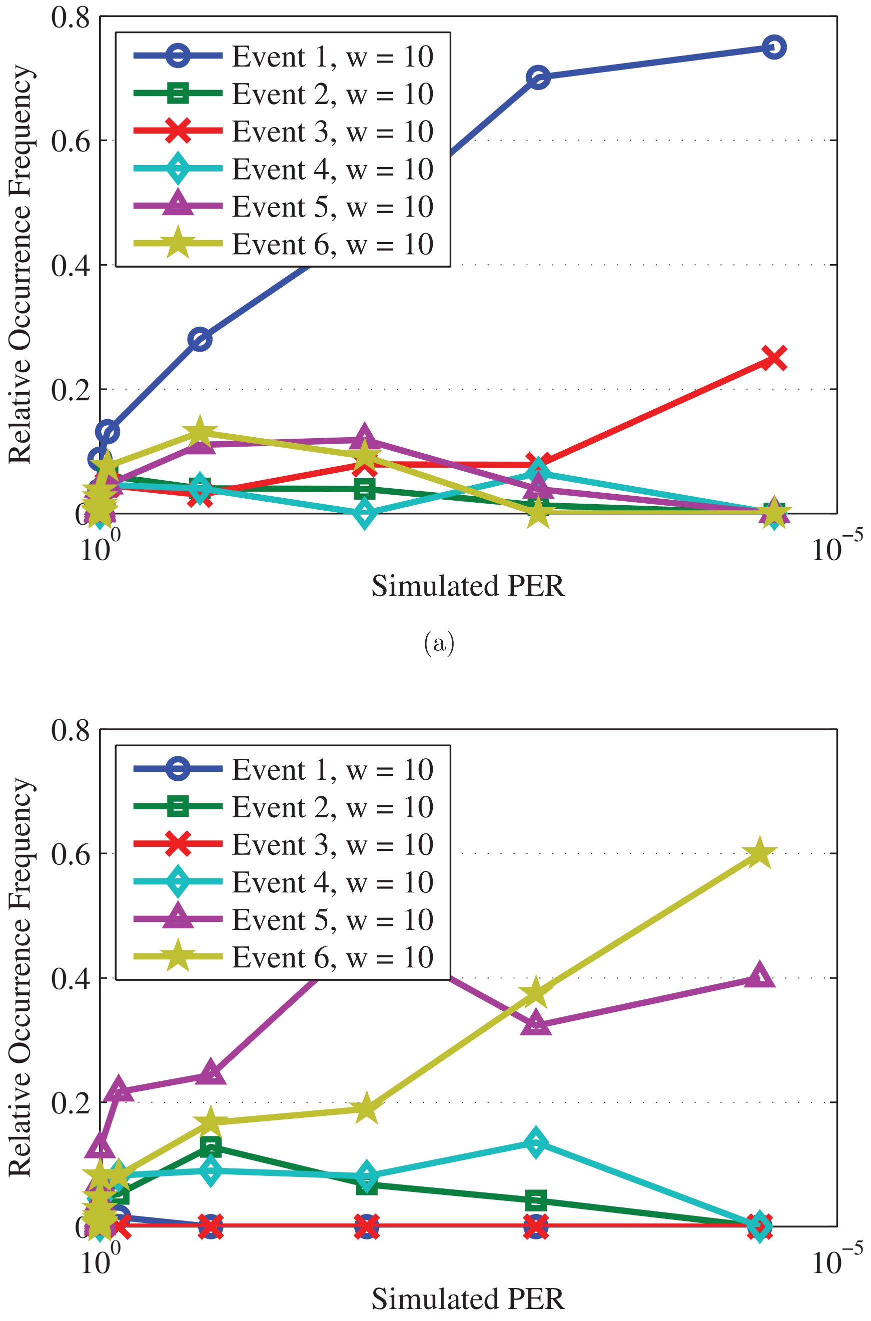
r	1	2	3	4
α_r	0.1453	0.1040	0.1047	0.0523
β_r	1.6212	0.7986	0.5581	0.5000

where $(D', \mathit{\Gamma}')$ has the same distribution as $\left(D'_j, \mathit{\Gamma}'_j\right)$ without the location information. The coefficients α_r and β_r of Gaussian Q-function approximation is given in Table 3.2. Finally, the PER of a system employing a uniform random interleaver can be calculated using (3.1) and (3.6).

3.3 Correlation of LLRs (HLELRs)

For a MIMO OFDM system employing a repeated-pattern interleaver, the LLRs (or HLELRs) cannot be treated as i.i.d. random variables. The reason is that, if the location of a bit in an OFDM symbol is known, its significance in a modulation symbol, the spatial stream and the subcarrier containing this bit, are all determined by the interleaver. This causes a huge impact on the PEP and can be demonstrated by the following example.

In the conventional analysis of (punctured) convolutional codes, the PEP $\mathrm{P}(\mathbf{e}_{t,l})$ is governed by the Hamming weight $w_{t,l}$ of the error event. This is valid only when a uniform random interleaver is used or when some specific system is operating in non-fading channels. To show a counter-example, we simulate a BPSK rate-1/2 system with one spatial stream (1x1) in TGn channel model B[Erc04] (having rms-delay spread $\tau_{\mathrm{rms}} = 15\mathrm{ns}$) and record the occurrence of each error event. Fig. 3.2 shows the simulated occurrence frequency of six error events having the same Hamming weight in two frequency selective channels at different SNR levels. Note that the six error events are differentiated by their trajectories t rather than their starting locations l. If a uniform random interleaver is used, we



(a)

(b)

Figure 3.2: The simulated occurrence frequency of six error events of a BPSK rate-1/2 1x1 OFDM system operating in two frequency selective channels (a) and (b) with $\tau_{\text{rms}} = 15$ns.

shall expect the six error events to have similar occurrence frequency. However, as we can see, the six error events are not equally probable. Furthermore, the two most probable error events (event 1 and 3) in the first channel realization are not likely to occur at all in the second channel realization. This shows that the occurrence frequency of an error event depends not only on its trajectory in the trellis diagram but also on the channel realization. Therefore, we should compute the PEP of each error event individually instead of relying on the transfer function[Vit71] or the weight spectrum of the code. However, it is not possible to individually deal with infinitely many error events specified by $\mathbf{T}$ in (3.1), so we only consider the error events in the finite set $\mathbf{T}'$, which is a subset of $\mathbf{T}$.

Typically, error events with low Hamming weights are likely to happen. Thus, let the set $\mathbf{T}'$ include all error events having the smallest four Hamming weights. For example, the smallest four Hamming weights of the error events of the well-known 64-state rate-1/2 $(133, 171)_8$ convolutional code are 10, 12, 14, and 16. For each starting location l, there are 1573 error events whose Hamming weights lie in this range. Since the PEP of an error event depends on the channel realization, an error event with a high Hamming weight could still be a dominant error event. Therefore, $\mathbf{T}'$ needs to contain some of the error events with higher Hamming weights discussed as below.

In the systems using high order QAM, the significance of bits is assigned by the interleaver in a round-robin manner; in the systems with multiple spatial streams, the interleaver maps the bit sequence to the spatial streams in a round-robin manner as well. These two facts are preferred while designing an interleaver to prevent bursty errors. However, they create an unintended high-low reliability pattern periodically applied to the bits. For example, in a 16-QAM system with the interleaver specified in [Sta09], each bit in an odd position before the interleaver is mapped to the first location of the in-phase or quadrature component. Hence, this bit is the most significant bit (MSB) and is more reliable. Let the period of the high-low reliability pattern be b bits. The value of b ranges from two to four because the highest order modulation scheme is 64-QAM (three bits per dimension) and the system allows up to four spatial streams. Although it is reasonable to set the maximum of b to twelve, the product of three and four, we set this number to four, the maximum of three and four, in order to reduce the computation complexity. We assume that low reliability appears once every b bits and the other bits has high reliability. Under this assumption, the finite set $\mathbf{T}'$ needs to contain all error events satisfying both of the following criteria.

- For each possible high-low reliability pattern, including its period b and the location of low reliability, we are interested in the error events that have the smallest five counts of bits with high reliability.
- For each count of bits with high reliability, we are interested in the error events that have the smallest five counts of bits with bad reliability.

In consideration of accuracy, we prefer $\mathbf{T}'$ to include as many error events as possible. However, the more error events are included, the higher complexity the method has. The way we pick the finite set $\mathbf{T}'$ achieves a good balance between accuracy and complexity in our computation environment. Note that, the output Hamming weight depends on the puncture state where the error event is initiated. Hence, the set $\mathbf{T}'$ should be calculated for each puncture state and is denoted as $\mathbf{T}'_l$ for $1 \leq l \leq M$, where M is the number of puncture states. If $l > M$, then $\mathbf{T}'_l = \mathbf{T}'_{(l-1 \mod M)+1}$.

Since the interleaver uses the same permutation pattern for all OFDM symbols, the error events having the same trajectory but starting from different OFDM symbols could possibly experience the same interleaving and have the same PEP. Thus, the PER expression in (3.1) can be simplified to

$$\mathrm{PER}_L \approx 1 - \prod_{l=1}^{\hat{L}} \left[1 - \sum_{t \in \mathbf{T}'_l} \mathrm{P}(\mathbf{e}_{t,l}) \right]^{L/\hat{L}}, \qquad (3.7)$$

where $\hat{L}$ is the period of the puncturer and the interleaver in terms of the number of information bits. It is given by

$$\hat{L} = \mathrm{lcm}(L_{\mathrm{OFDM}}, L_{\mathrm{punc.}}) \frac{M}{L_{\mathrm{punc.}}}, \tag{3.8}$$

where lcm(·) calculates the least common multiple, L_{OFDM} is the number of bits in an OFDM symbol, $L_{\mathrm{punc.}}$ is the total number of puncturer output bits of a puncture period, and M is the number of puncture states equal to the total number of information bits of a puncture period. In the following, we will present the calculation of the PEP in (3.3) using the HLELRs with the consideration of their correlation.

The correlation of bits' HLELRs can be separated into three categories according to the locations of the bits: a) the bits come from different subcarriers (correlation across subcarriers), b) the bits come from the same subcarrier but from different spatial streams (correlation across spatial streams), and c) the bits come from the same subcarrier and the same spatial stream (correlation within a subcarrier).

3.3.1 Correlation across Subcarriers

To demonstrate the impact of correlation across subcarriers, we consider a BPSK rate-1/2 system with one spatial stream operating in TGn channel model B ($\tau_{\mathrm{rms}} =$ 15ns). Table 3.3 shows the permutation of the interleaver. The bits are filled in along the rows and taken out along the columns. The order of the interleaver output determines the subcarrier where a bit goes. For example, the first bit goes to the first subcarrier, the fourteenth bit goes to the second subcarrier, and the second bit goes to the fifth subcarrier. In this table, the highlighted locations are the bits associated with a specific error event. It is clear that nine out of ten bits associated with this error event are mapped to subcarriers close to each other. These subcarriers may have similar channel gains because of the coherence band-

Table 3.3: The interleaving of a BPSK system with one spatial stream. The highlighted locations are the bits associated with a specific error event.

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	$\boxed{17}$	$\boxed{18}$	$\boxed{19}$	20	$\boxed{21}$	22	23	24	25	26
27	28	29	30	$\boxed{31}$	32	33	34	$\boxed{35}$	36	37	38	39
$\boxed{40}$	41	42	43	44	$\boxed{45}$	46	$\boxed{47}$	$\boxed{48}$	59	50	51	52

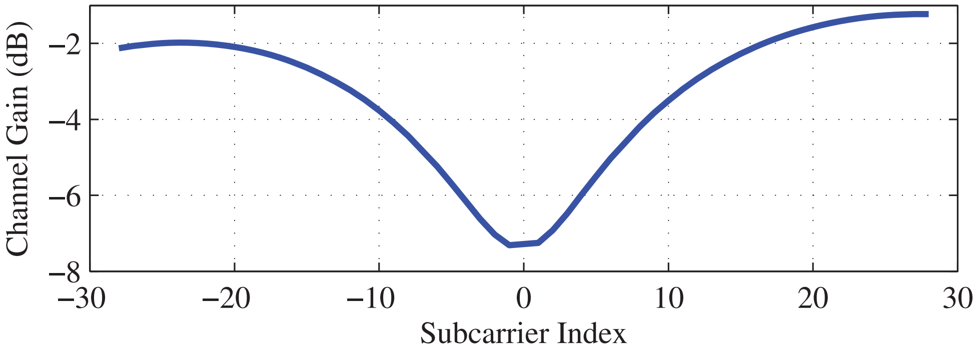


Figure 3.3: The frequency response of a frequency selective channel with $\tau_{\text{rms}} =$ 15ns.

width. Fig. 3.3 illustrates the frequency response of a specific channel realization with $\tau_{\text{rms}} = 15$ns. We observe that the error event whose bits are highlighted in Table 3.3 is more likely to happen than most of the other error events having the same Hamming weight. The reason is that most of the bits associated with this error event are experiencing the fade of the channel and hence have low PPSNRs.

To model this effect, we need to treat the PPSNRs $\mathit{\Gamma}_i$ in (3.4) and (3.5) as deterministic numbers instead of random variables. Furthermore, since the significance of a bit is fixed even in a high order QAM system, the parameter pair $(D_i, \mathit{\Phi}_i)$ is either $(D_{\text{I},1}, \mathit{\Phi}_{\text{I},1})$, $(D_{\text{I},2}, \mathit{\Phi}_{\text{I},2})$, or $(D_{\text{I},3}, \mathit{\Phi}_{\text{I},3})$, depending on the bit's significance. However, $(D_i, \mathit{\Phi}_i)$ is still a pair of random variables in general and governed by the transmitted constellation point. For example, according to Table 3.1, if a bit is the least significant bit (LSB) of a 16-QAM symbol, its parameter pair $(D_i, \mathit{\Phi}_i)$

is either $(1,1)$ or $(1,-3)$ with equal probability. Thus, similar to (3.5), the conditional distribution of $Y_i^{\star\star}$ given only the parameter pair $(D_i, \varPhi_i)$ follows a normal distribution

$$Y_i^{\star\star}|(D_i = d_i, \varPhi_i = \phi_i) \sim \mathcal{N}\left(-\left[d_i + \mathrm{d}_{\mathrm{adj}}\left(\frac{d_i\varGamma_i}{-g}, \frac{\varGamma_i}{g}, \frac{\phi_i\varGamma_i}{g}\right)\right]\frac{\varGamma_i}{g}, \frac{\varGamma_i}{g}\right) \tag{3.9}$$

because $\varGamma_i$ is deterministic.

Since the HLELR $Y_i^\star$ is approximated by $Y_i^{\star\star}$, the PEP in (3.3) is approximated as

$$\mathrm{P}(\mathbf{e}_{t,l}) \approx \mathrm{P}\left(\sum_{j=1}^{w_{t,l}} Y_{\mathrm{i}(t,l,j)}^{\star\star} > 0\right). \tag{3.10}$$

Using the Gaussian Q-function approximation[CDS03], we approximate the PEP as

$$\mathrm{P}(\mathbf{e}_{t,l}) \approx \sum_{r=1}^{4} \alpha_r \,\mathrm{E}\left[\exp\left(\sum_{j=1}^{w_{t,l}} -\beta_r \varGamma_j' \left\{D_j' + \left(D_j' - 1\right)\mathrm{E}\left[D'\right]\right\}\right)\right], \tag{3.11}$$

where the parameters α_r and β_r are listed in Table 3.2, and the outer expectation is taken over all D_j'.

To show the importance of the consideration of correlation across subcarriers, we simulate the above BPSK rate-1/2 system with one spatial stream in several frequency selective channels generated from TGn channel model B ($\tau_{\mathrm{rms}} = 15\mathrm{ns}$). We then compare the PER predictions from (3.11), which takes into account the correlation across subcarriers, with the PER predictions from (3.6), which assumes a uniform random interleaver while the simulated system uses a repeated-pattern interleaver. The metric we use to compare the prediction methods is the mean square error (MSE) defined as

$$\mathrm{MSE} = \frac{1}{N}\sum_{n=1}^{N} \left|\ln\left(\mathrm{PER}_{\mathrm{pred.},n}\right) - \ln\left(\mathrm{PER}_{\mathrm{sim.},n}\right)\right|^2, \tag{3.12}$$

where N is the total number of simulation points, and $\mathrm{PER}_{\mathrm{pred.},n}$ and $\mathrm{PER}_{\mathrm{sim.},n}$ are the predicted and simulated PER values, respectively, for the n^{th} channel realization.

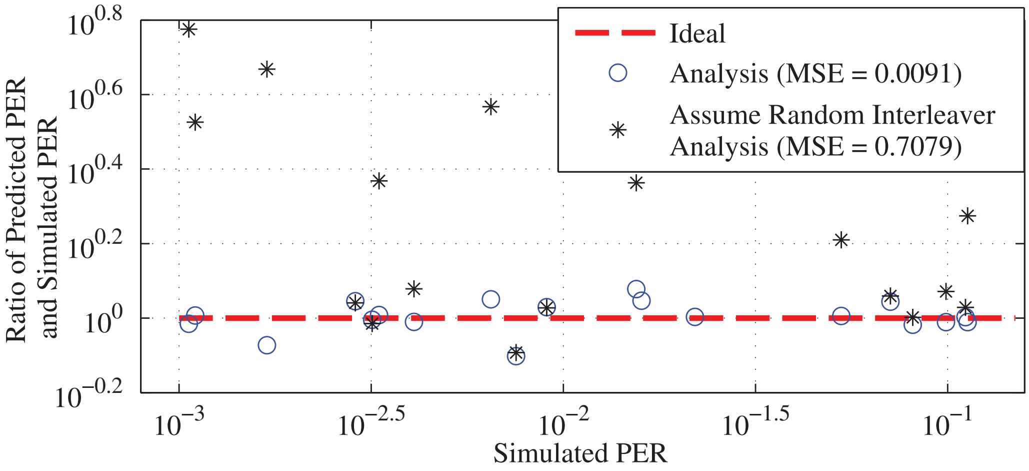


Figure 3.4: Accuracy of PER prediction for BPSK rate-1/2 scheme with packet length = 1024 bytes in 1x1 frequency selective channels generated according to TGn channel model B ($\tau_{\text{rms}} = 15$ns). A vertical set of scattered points corresponds to one channel realization. The mean square error is calculated by (3.12).

In Fig. 3.4, the method that assumes a uniform random interleaver tends to over-estimate the PER in some of the channel realizations and has an MSE of 0.7079. To the contrary, the method that takes into account the correlation across subcarriers has very high accuracy and its MSE is only 0.0091. Therefore, it is important to consider the correlation across subcarriers while predicting the PER. Since the simulated system is a BPSK system with one spatial stream, there is no correlation across spatial streams or correlation within a subcarrier discussed in the following subsections.

3.3.2 Correlation across Spatial Streams

The correlation across spatial streams comes from two or more bits mapped to the same subcarrier but different spatial streams. Consider a 2x2 system with two spatial streams and a ZF MIMO decoder at the receiver. Let the noise of a subcarrier at the two antennas be N_1 and N_2. Also, let $u_{i,j}$ be the coefficient of the

ZF MIMO decoder for the i^{th} spatial stream and the j^{th} antenna. The effective noise of the two spatial streams after the MIMO decoder is given by

$$\begin{bmatrix} \hat{N}_1 \\ \hat{N}_2 \end{bmatrix} = \begin{bmatrix} u_{1,1} & u_{1,2} \\ u_{2,1} & u_{2,2} \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} .$$

Although N_1 and N_2 are independent zero-mean circularly-symmetric complex normal random variables, $\hat{N}_1$ and $\hat{N}_2$ are correlated in general. Assume an HLELR is a linear operation of the corresponding received modulation symbol. Two HLELRs will be correlated because their noise components are correlated.

Since the variance of N_1 and N_2 on each dimension is $N_0/2$, the variance of $\hat{N}_1$ and $\hat{N}_2$ on each dimension is $\left(|u_{1,1}|^2 + |u_{1,2}|^2\right) N_0/2$ and $\left(|u_{2,1}|^2 + |u_{2,2}|^2\right) N_0/2$, respectively. If two of the bits associated with an error event come from the two modulation symbols affected by $\hat{N}_1$ and $\hat{N}_2$ respectively, then the sum of their effective noise, which occurs in the sum of their HLELRs, has variance of

$$\left(|C_1 u_{1,1} + C_2 u_{2,1}|^2 + |C_1 u_{1,2} + C_2 u_{2,2}|^2\right) N_0/2 \ . \tag{3.13}$$

For $i \in \{1, 2\}$, the random variable $C_i \in \{1, -1\}$ if the bit from the i^{th} spatial stream belongs to the in-phase component, and $C_i \in \left\{\sqrt{-1}, -\sqrt{-1}\right\}$ otherwise. The sign of C_i, determining if the noise is combined constructively or destructively, is governed by the transmitted bit being 0 or 1, so the two possibilities are equally probable. If the HLELRs are treated as independent random variables, the variance of the sum of the effective noise is

$$\left(|u_{1,1}|^2 + |u_{2,1}|^2 + |u_{1,2}|^2 + |u_{2,2}|^2\right) N_0/2 \ . \tag{3.14}$$

Compared to (3.13), this value is under-estimated by

$$\left[\Re\left(C_1 u_{1,1} C_2^* u_{2,1}^*\right) + \Re\left(C_1 u_{1,2} C_2^* u_{2,2}^*\right)\right] N_0, \tag{3.15}$$

where $\Re(\cdot)$ returns the real part of the complex number and the superscript "*" is the complex conjugate operator.

To characterize the effect of the correlation across spatial streams, we should modify the PEP in (3.11) by including a correction term of the variance. Previously, we approximated the conditional distribution of the sum of HLELRs given the transmitted sequence as a normal distribution. Now its mean is still $\sum_{j=1}^{w_{t,l}} D'_j \mathit{\Gamma}'_j$ but its variance becomes $\sum_{j=1}^{w_{t,l}} \mathit{\Gamma}'_j + \sum_{f=1}^{F} W_f$, where F is the total number of subcarriers and W_f is the additional variance, such as (3.15), caused by the correlation across spatial streams in the f^{th} subcarrier and associated with the pairwise error event $\mathbf{e}_{t,l}$. Let $w_{t,l,f}$ be the number of bits that are associated with the pairwise error event $\mathbf{e}_{t,l}$ and belong to the f^{th} subcarrier. If only one or no bits in the f^{th} subcarrier are associated with this error event, i.e., $w_{t,l,f} < 2$, then $W_f = 0$. When $w_{t,l,f} \geq 2$, W_f is given by

$$W_f = \sum_{j=1}^{N_{\mathrm{R}}} \left(\left| \sum_{i=1}^{w_{t,l,f}} C'_i u'_{i,j} \right|^2 - \sum_{i=1}^{w_{t,l,f}} \left| u'_{i,j} \right|^2 \right) \frac{N_0}{2}, \tag{3.16}$$

where N_{R} is the number of receive antennas. The random variables C'_i is similar to C_i defined above but the subscript represents the index of the bit in this subcarrier instead of the index of the spatial stream. Similarly, the coefficient $u'_{i,j}$ is the coefficient of the ZF MIMO decoder for the i^{th} bit in this subcarrier and the j^{th} receive antenna.

With the knowledge of the mean and variance, we can calculate the PEP as

$$\mathrm{P}(\mathbf{e}_{t,l}) \approx \mathrm{E}\left[\mathrm{Q}\left(\sqrt{ \sum_{j=1}^{w_{t,l}} \mathit{\Gamma}'_j \left\{ D'_j + \left(D'_j - 1 \right) \mathrm{E}[D'] \right\} - \sum_{f=1}^{F} W_f \, \mathrm{E}[D'] } \right) \right], \tag{3.17}$$

where the outer expectation is taken over all $\left(D'_j, W_f \right)$. Note that D'_j and W_f are both governed by the transmitted bit sequence. Applying the Gaussian Q-function approximation to (3.17) yields the PEP. The PER is then calculated by (3.7).

To illustrate the importance of the consideration of correlation across spatial streams, we simulate a BPSK rate-5/6 system with three spatial streams in several

Table 3.4: The interleaving of a BPSK system with one spatial stream. The highlighted locations are the bits associated with a specific error event.

	1	4	7	10	13	16	19	22	25	28	31	34	37
1st	40	$\boxed{43}$	$\boxed{46}$	49	52	55	58	61	64	67	70	73	76
Str.	79	82	85	88	91	94	97	100	103	106	109	112	115
	118	121	124	127	130	133	136	139	142	145	148	151	154
	95	98	101	104	107	110	113	116	80	83	86	89	92
2nd	134	137	140	143	146	149	152	155	119	122	125	128	131
Str.	20	23	26	29	32	35	38	2	5	8	11	14	17
	59	62	65	68	71	74	77	41	44	47	50	53	56
	126	129	132	135	138	141	144	147	150	153	156	120	123
3rd	12	$\boxed{15}$	$\boxed{18}$	21	24	27	30	33	36	39	3	6	9
Str.	51	54	57	60	63	66	69	72	75	78	42	45	48
	90	93	96	99	102	105	108	111	114	117	81	84	87

flat fading channels generated from TGn channel model A ($\tau_{\text{rms}} = 0\text{ns}$). We then compare the PER prediction that considers the correlation across subcarriers and spatial streams through (3.17) with the PER prediction that only considers the correlation across subcarriers through (3.11) but assumes independent HLELRs. The permutation of the interleaver is shown in Table 3.4, where the highlighted locations are the bits associated with a specific error event. For this error event, the total four associated bits belong to only two subcarriers from two spatial streams. Hence, the correlation across spatial streams could possibly cause a huge impact on the PEP of this error event. If the channel realization happens to make W_f a big number, we shall expect to see a big prediction error for the method that assumes independent HLELRs.

In Fig. 3.5, the method that assumes independent HLELRs has predicted PERs far below the simulated PERs in some of the channel realizations and its

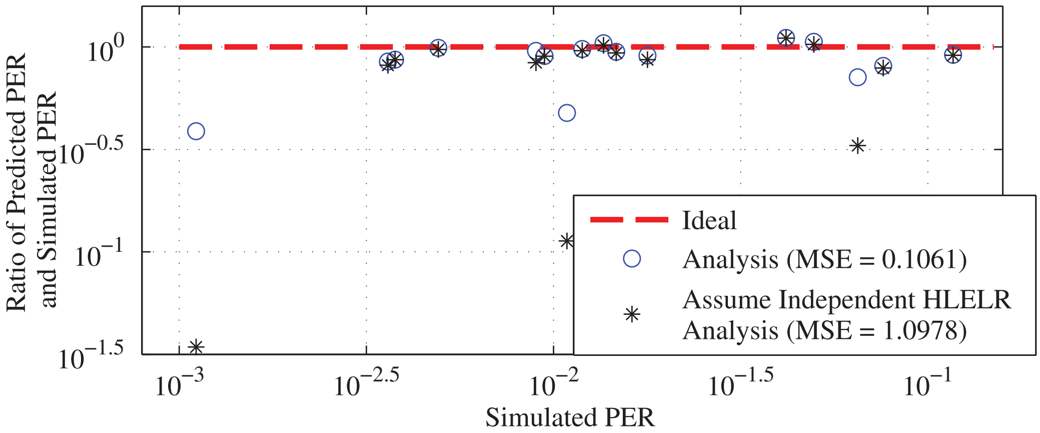


Figure 3.5: Accuracy of PER prediction for BPSK rate-5/6 scheme with packet length = 1024 bytes in 3x3 flat fading channels generated according to TGn channel model A (τ_{rms} = 0ns). A vertical set of scattered points corresponds to one channel realization. The mean square error is calculated by (3.12).

MSE is 1.0978. Note that, this method does consider the correlation across subcarriers, which accounts for PPSNR differences among spatial streams. The MSE of the method that takes into account the correlation across spatial streams is 0.1061. Thus, considering the correlation across spatial streams is crucial in the PER prediction. Although the proposed method performs a lot better than the method that assumes independent HLELRs, there is still room for improvement in certain channels. Since the simulated system uses BPSK modulation, there is no correlation within a subcarrier discussed in the next subsection.

3.3.3 Correlation within a Subcarrier

The correlation within a subcarrier comes from two or more bits mapped to the same subcarrier and the same spatial stream. These bits will experience the same channel gain. Moreover, if two bits belong to the same in-phase component or quadrature component, they will observe exactly the same noise. Therefore, when we calculate the sum of their HLELRs, the sum of their noise could be doubled or totally canceled.

For example, consider the two bits, $X_{\mathrm{I},1}$ and $X_{\mathrm{I},2}$, of the in-phase component of a 16-QAM symbol. Let Z_{I} be the scaled in-phase component of the received symbol and it is given by

$$Z_{\mathrm{I}} = \mathrm{sgn}(X_{\mathrm{I},1})\,[2 - \mathrm{sgn}(X_{\mathrm{I},2})] + N_{\mathrm{I}}\sqrt{2g/E_{\mathrm{s}}},$$

where N_{I} is the in-phase component of the noise, g is a modulation dependent scaling factor given in Table 3.1, and E_{s} is the received symbol energy. According to the simplified demapper[TB02], the LLRs of the MSB and the LSB are

$$\begin{cases} Y_{\mathrm{I},1} = -2Z_{\mathrm{I}}\mathit{\Gamma}/g \\ Y_{\mathrm{I},2} = -2\left(-\left|Z_{\mathrm{I}}\right| + 2\right)\mathit{\Gamma}/g \end{cases},$$

where $\mathit{\Gamma}$ is the PPSNR or this symbol. Assume that $(X_{\mathrm{I},1}, X_{\mathrm{I},2}) = (0, 0)$ is transmitted. Based on (3.2), the corresponding HLELRs are

$$\begin{cases} Y^{\star}_{\mathrm{I},1} = \left(-3 + N_{\mathrm{I}}\sqrt{2g/E_{\mathrm{s}}}\right)\mathit{\Gamma}\Big/g \\ Y^{\star}_{\mathrm{I},2} = \left(-\left|-3 + N_{\mathrm{I}}\sqrt{2g/E_{\mathrm{s}}}\right| + 2\right)\mathit{\Gamma}\Big/g \end{cases}.$$

Depending on the realization of the noise, the sum of their HLELRs is

$$Y^{\star}_{\mathrm{I},1} + Y^{\star}_{\mathrm{I},2} = \begin{cases} \left(-4 + 2N_{\mathrm{I}}\sqrt{2g/E_{\mathrm{s}}}\right)\mathit{\Gamma}\Big/g & \text{if} - 3 + N_{\mathrm{I}}\sqrt{2g/E_{\mathrm{s}}} \leq 0 \\ 2\mathit{\Gamma}/g & \text{otherwise} \end{cases}.$$

The noise is doubled in the first case, and it is totally canceled in the second case when the realization of the noise is a relatively big positive number, which is not likely to happen in a high SNR region. The distribution of their sum is a normal distribution with one tail replaced with a point mass of probability. We can also treat this distribution as a piecewise normal distribution while one of the segment has zero variance. The same result can be obtained if $(X_{\mathrm{I},1}, X_{\mathrm{I},2}) = (1, 1)$. Under the same setting, assume that $(X_{\mathrm{I},1}, X_{\mathrm{I},2}) = (0, 1)$ is sent. The sum of their

HLELRs is

$$Y^{\star}_{\mathrm{I},1} + Y^{\star}_{\mathrm{I},2} = \begin{cases} -2\mathit{\Gamma}/g & \text{if } -1 + N_{\mathrm{I}}\sqrt{2g/E_{\mathrm{s}}} \leq 0 \\ \left(-4 + 2N_{\mathrm{I}}\sqrt{2g/E_{\mathrm{s}}}\right) \mathit{\Gamma} \Big/ g & \text{otherwise} \end{cases} .$$

Similarly, the noise is doubled in one case and totally canceled in the other case. However, the case where the noise is totally canceled is more likely to happen now. The distribution of their sum is a point mass of probability with one tail of a normal distribution. The same result can be obtained if $(X_{\mathrm{I},1}, X_{\mathrm{I},2}) = (1, 0)$.

The above is an example of a 16-QAM symbol. Similar results can be obtained in a 64-QAM symbol. Hence, the conditional distribution of the sum of their HLELRs, given the transmitted constellation, follows piecewise normal distribution in general. This distribution can be approximated as a normal distribution using the method described in Section 2.B.

To characterize the effect of the correlation within a subcarrier, we shall group the HLELRs which belong to the same in-phase or quadrature component of a subcarrier and a spatial stream, and compute the distribution of their sum. Let $\nu_{t,l}$ be the total number of such groups in the bits associated with the pairwise error event $\mathbf{e}_{t,l}$. Note that $\nu_{t,l}$ depends on not only the error event but also the interleaver. For the k^{th} group of the pairwise error event $\mathbf{e}_{t,l}$, we approximate the conditional distribution of the sum of their HLELRs, given the transmitted constellation, using a normal distribution with adjusted mean given in Appendix 2.A. The variance and mean of this normal distribution are $\mathit{\Gamma}^{\star}_k$ and $D^{\star}_k \mathit{\Gamma}^{\star}_k$, respectively, where $D^{\star}_k$ and $\mathit{\Gamma}^{\star}_k$ are both random variables governed by the transmitted constellation.

Therefore, with the Gaussian Q-function approximation, the PEP that includes all three types of correlation is given by

$$\mathrm{P}(\mathbf{e}_{t,l}) \approx \sum_{r=1}^{4} \alpha_r \, \mathrm{E}\Bigg[\exp\Bigg\{ -\beta_r \Bigg(\sum_{j \in \mathbf{w}_{t,l}} \mathit{\Gamma}'_j \left\{ D'_j + \left(D'_j - 1\right) \mathrm{E}[D'] \right\} - \sum_{f=1}^{F} W_f \, \mathrm{E}[D']$$

Table 3.5: Half of the interleaving of a 16-QAM system with one spatial stream. The highlighted locations are the bits associated with a specific error event.

I_1	$\boxed{1}$	$\boxed{15}$	$\boxed{3}$	17	5	19	7	$\boxed{21}$	9	$\boxed{23}$	$\boxed{11}$	25	13
I_2	14	$\boxed{2}$	16	4	18	$\boxed{6}$	20	$\boxed{8}$	22	10	$\boxed{24}$	12	26
Q_1	27	41	29	43	31	45	33	47	35	49	37	51	39
Q_2	40	28	42	30	44	32	46	34	48	36	50	38	52
I_1	53	67	55	69	57	71	59	73	61	75	63	77	65
I_2	66	54	68	56	70	58	72	60	74	62	76	64	78
Q_1	79	93	81	95	83	97	85	99	87	101	89	103	91
Q_2	92	80	94	82	96	84	98	86	100	88	102	90	104

$$+\sum_{k=1}^{\nu_{t,l}} \mathit{\Gamma}_k^\star \left\{D_k^\star + (D_k^\star - 1)\,\mathrm{E}[D']\right\}\Bigg)\Bigg\}\Bigg], \tag{3.18}$$

where $\mathbf{w}_{t,l}$ is the set containing the indices of the bits that are associated with the pairwise error event $\mathbf{e}_{t,l}$ but do not belong to any of the $\nu_{t,l}$ groups. Note that, the outer expectation is taken over all the random variables (D'_j,W_f, $\mathit{\Gamma}_k^\star$, and $D_k^\star$), which are all governed by the transmitted bit sequence. Finally, the PER can be calculated using (3.7) and (3.18).

To show the importance of the consideration of correlation within a subcarrier, we simulate a 16-QAM rate-1/2 system with one spatial stream in several flat fading channels generated from TGn channel model A ($\tau_{\mathrm{rms}} = 0\mathrm{ns}$). We then compare the PER prediction that considers all three types of correlation through (3.18) with the PER prediction that only considers the correlation across subcarriers through (3.11) but assumes the HLELRs are independent. Note that, since the simulated system has only one spatial stream, the correlation across spatial streams does not exist. The permutation of the interleaver is shown in Table 3.5, where the highlighted locations are the bits associated with a specific error event. In this error event, there are three associated bit pairs satisfying that each of them

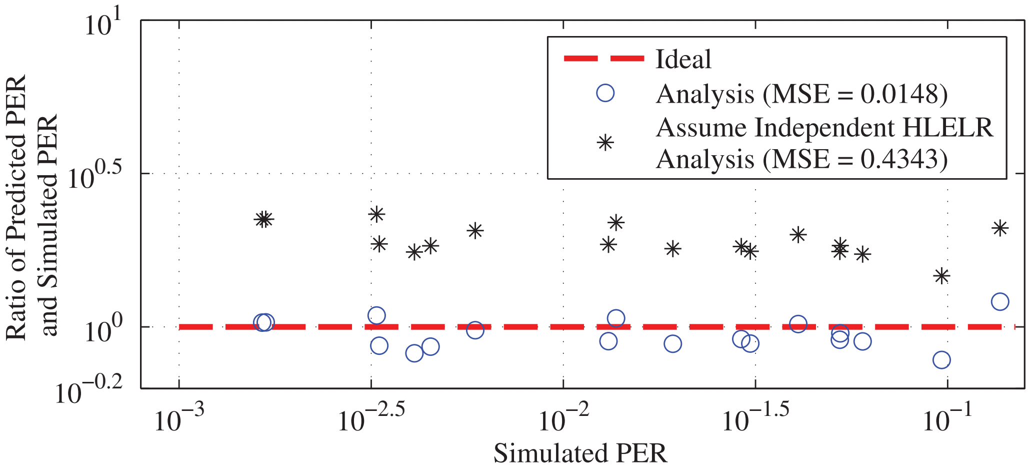


Figure 3.6: Accuracy of PER prediction for 16-QAM rate-1/2 scheme with packet length = 1024 bytes in 1x1 flat fading channels generated according to TGn channel model A ($\tau_{\text{rms}} = 0$ns). A vertical set of scattered points corresponds to one channel realization. The mean square error is calculated by (3.12).

belongs to the in-phase component of a subcarrier. Hence, the PEP of this error event is greatly impacted by the correlation within a subcarrier.

In Fig. 3.6, the method that assumes independent HLELRs has an MSE of 0.4343. Although this method does consider the correlation across subcarriers, which includes the significance of each bit in a 16-QAM symbol, the lack of modeling the correlation within a subcarrier makes it over-estimate the PERs in all of the channel realizations. The prediction results of the method that considers all three types of correlation are scattered around the ideal line and its MSE is 0.0148. This huge improvement is achieved by solely considering the correlation within a subcarrier.

3.4 Numerical Result

In this section, we compare the proposed PER prediction method with the existing methods by looking at their accuracy in various transmission schemes. The MSE between the predicted PERs and the simulated PERs is the performance metric and calculated using (3.12). The averaged MSE is the linear average of the MSEs of all transmission schemes and can be seen as an overall metric. The simulated MIMO OFDM system follows the IEEE 802.11n standard[Sta09] with packet length set to 1024 bytes. The simulated frequency selective channels are generated based on the TGn channel model A through F[Erc04], which has an rms-delay spread τ_{rms} up to 150ns.

In addition to the proposed analytical method referred to as the Q-function approximation method, there are five other PER prediction methods mentioned in this section. The saddlepoint approximation method is an analytical approach discussed in Section 2.3.2. This method relies on the saddlepoint approximation [MFC06] to calculate the tail probability of the sum of i.i.d. random variables. Thus, this method works well for systems with a uniform random interleaver, where the HLELRs are i.i.d. random variables, but not a repeated-pattern interleaver.

The reset of the methods are all heuristic approaches. They are cumulant generating function based effective SNR mapping (κESM)[SGL09], exponential effective SNR mapping (EESM)[Eri03], mutual information effective SNR mapping (MIESM)[TS03], and mean mutual information per coded bit mapping (MMIBM) [JKW10]. They average the PPSNRs through certain functions with calibrated parameters to get a single metric that is capable of representing the quality of the link. Then, a look-up table generated by simulations in additive white-Gaussian noise (AWGN) channels is used to map from the single metric to the PER. Note that the calibrated parameter and look-up table should be prepared for each trans-

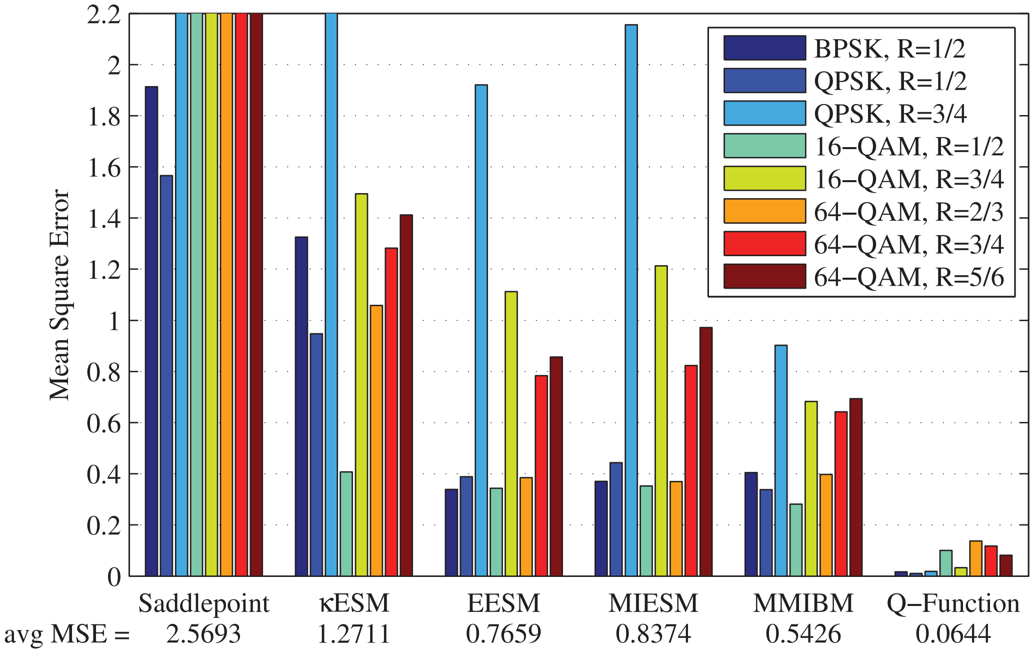


Figure 3.7: Mean square error of PER prediction for various transmission schemes with one spatial stream and packet length = 1024 bytes in 1x1 frequency selective channels having an rms-delay spread τ_{rms} up to 150ns. The averaged MSE is the linear average of the MSEs of all transmission schemes.

mission scheme, which is the combination of modulation scheme, code rate, and number of spatial streams.

Fig. 3.7 shows the accuracy of the PER prediction methods for systems with one spatial stream. In this case, the correlation across spatial streams does not exist but the other two types of correlation may exist. The saddlepoint approximation method assumes a uniform random interleaver and has a relatively high averaged MSE of 2.5693. The heuristic methods have moderate averaged MSEs ranging from 0.5426 to 1.2711. The averaged MSE of the Q-function approximation method is 0.0644, which is much more accurate than all the other methods. Note that none of the transmission schemes predicted by the Q-function approximation method has an MSE higher than 0.2.

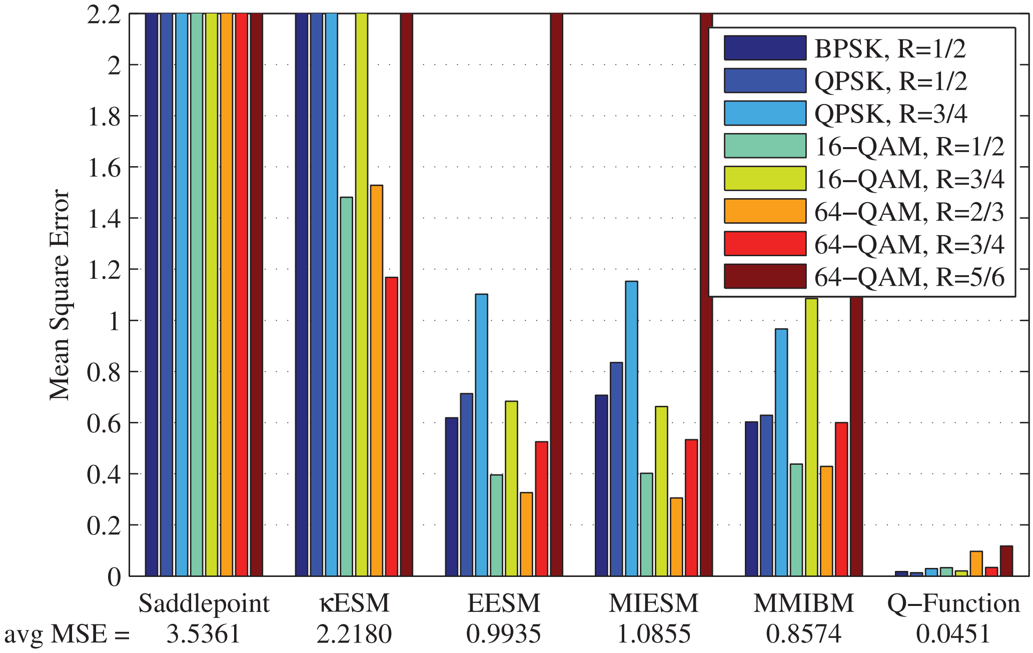


Figure 3.8: Mean square error of PER prediction for various transmission schemes with two spatial streams, packet length = 1024 bytes, and ZF in 2x2 frequency selective channels having an rms-delay spread τ_{rms} up to 150ns. The averaged MSE is the linear average of the MSEs of all transmission schemes.

Fig. 3.8 illustrates the accuracy of the PER prediction methods for systems with two spatial streams. Under this setting, all three types of correlation may exist. The Q-function approximation method has an averaged MSE of 0.0451 and outperforms all the other methods. Similar trends can be seen in Fig. 3.9 and Fig. 3.10, which show the accuracy of the PER prediction methods for systems with three spatial streams and four spatial streams, respectively. In general, the Q-function approximation method has about 10% of the averaged MSE of the best existing method.

In Fig. 3.10, we observe that the Q-function approximation method has relatively high MSEs for two transmission schemes, compared to the other schemes. This could origin from the fact that the interleavers of these two schemes create

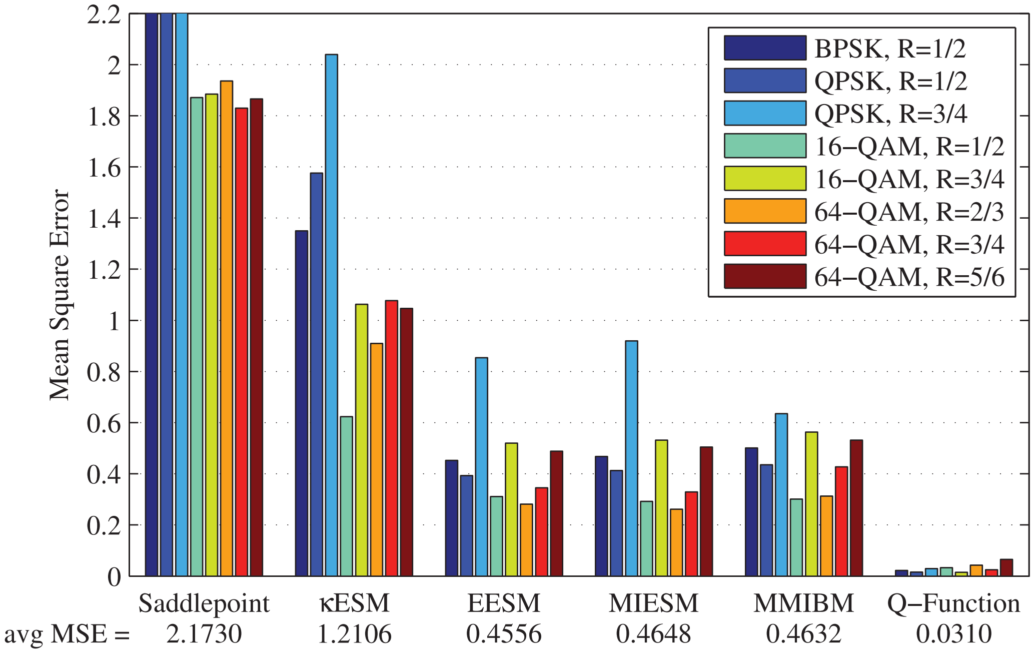


Figure 3.9: Mean square error of PER prediction for various transmission schemes with three spatial streams, packet length = 1024 bytes, and ZF in 3x3 frequency selective channels having an rms-delay spread τ_{rms} up to 150ns. The averaged MSE is the linear average of the MSEs of all transmission schemes.

certain HLELR correlation that has not been properly characterized yet. Nevertheless, these MSEs are still much smaller than the MSEs of these two schemes predicted by the other existing methods.

In the heuristic methods, κESM has no calibrated parameters, EESM and MIESM use one calibrated parameter each, and MMIBM requires two calibrated parameters. This explains why κESM performs worst and MMIBM is the best among the heuristic methods.

In the previous chapter, we have seen that the heuristic methods have comparable accuracy with the Q-function approximation method for the systems employing a uniform random interleaver. However, the Q-function approximation method is far more accurate than the heuristic methods for the systems using

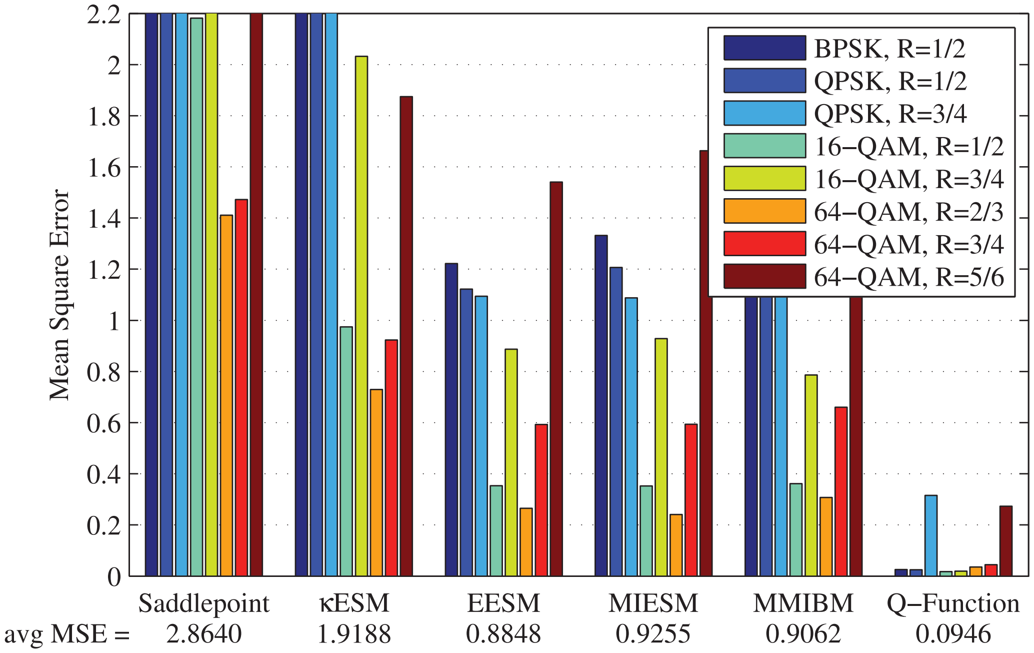


Figure 3.10: Mean square error of PER prediction for various transmission schemes with four spatial streams, packet length = 1024 bytes, and ZF in 4x4 frequency selective channels having an rms-delay spread τ_{rms} up to 150ns. The averaged MSE is the linear average of the MSEs of all transmission schemes.

a repeated-pattern interleaver. One of the reasons is that the heuristic methods only use the distribution of the PPSNRs but not the order (locations) of them in a frequency selective channel. Another reason is that the heuristic methods rely on the simulated PERs in the AWGN channel to map from the single metric to the PER. However, in the AWGN channel, the correlation across subcarriers cannot be seen because when all subcarriers have the same channel gain, their order does not matter.

3.5 Conclusion

In this chapter, we describe an accurate analytical PER prediction method for punctured convolutionally coded MIMO OFDM systems with a repeated-pattern interleaver. We begin with the Q-function approximation method for systems with a uniform random interleaver, and generalize it to be capable of predicting the PERs of systems with a repeated-pattern interleaver. The HLELRs of such systems are not i.i.d. random variables but have correlation brought about by the interleaver. This correlation can be divided into three categories: a) correlation across subcarriers, b) correlation across subcarriers, and c) correlation within a subcarrier. With the consideration of the three types of correlation, the Q-function approximation method delivers high prediction accuracy for all combinations of modulation scheme, code rate, and number of spatial streams in a wide range of MIMO frequency selective channels. Furthermore, the proposed method outperforms all the other existing PER prediction methods and its averaged MSE is about 10% of the averaged MSE of the best existing method.

CHAPTER 4

Convolutional-Code Specific CRC Code Design to Minimize the Probability of Undetected Error

Cyclic redundancy check (CRC) codes check if a codeword is correctly received. This chapter presents an algorithm to design CRC codes that are optimized for the code-specific error behavior of a specified feedforward convolutional code. The algorithm utilizes two distinct approaches to computing undetected error probability of a CRC code used with a specific convolutional code. The first approach enumerates the error patterns of the convolutional code and tests if each of them is detectable. The second approach reduces complexity significantly by exploiting the equivalence of the undetected error probability to the frame error rate of an equivalent catastrophic convolutional code. The error events of the equivalent convolutional code are exactly the undetectable errors for the original concatenation of CRC and convolutional codes. This simplifies the computation because error patterns do not need to be individually checked for detectability. As an example, we optimize CRC codes for a commonly used 64-state convolutional code for information length k=1024 demonstrating significant reduction in undetected error probability compared to the existing CRC codes with the same degrees. For a fixed target undetected error probability, the optimized CRC codes typically require two fewer bits.

This chapter is organized as follows: Section 4.1 provides the system model. Section 4.2 presents the exclusion and construction methods for computing the undetected error probability of a CRC code concatenated with an inner convolu-

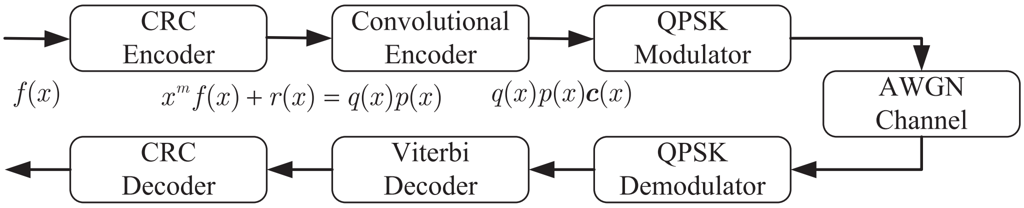


Figure 4.1: Block diagram of a system employing CRC and convolutional codes.

tional code. Section 4.3 describes how these two methods can be used to design a CRC code to minimize the undetected error probability for a specific feedforward convolutional code and information length. Section 4.4 applies this design approach to the most common 64-state convolutional code for information length $k = 1024$. Section 4.5 concludes this chapter.

4.1 System Model

Fig. 4.1 shows the block diagram of our system model employing a CRC code concatenated with a convolutional code in an additive white Gaussian noise (AWGN) channel. The k-bit information sequence is expressed as a binary polynomial $f(x)$ of degree smaller than or equal to $k-1$, and determined by the information bits: the first bit determines the coefficient of x^{k-1}, the second bit determines the coefficient of x^{k-2}, and so on. At the CRC encoder output, m parity bits, which are determined by the remainder $r(x)$ of $x^m f(x)$ divided by the degree-m CRC generator polynomial $p(x)$, are appended to the information sequence. Thus the $n = k + m$-bit sequence described by $x^m f(x) + r(x)$ is divisible by $p(x)$ producing the k-bit quotient $q(x)$ and a remainder of zero. The CRC-encoded sequence can also be expressed as $q(x)p(x)$. Thus $q(x)$ has a one-to-one relationship with $f(x)$. Note that $q(x)p(x)$ is the result of processing the sequence $x^m q(x)$ ($q(x)$ and m trailing zeros) by the CRC encoder circuit described by $p(x)$.

The transmitter uses a feedforward, terminated, rate-$\frac{1}{N}$ convolutional code having ν memories with generator polynomial $\boldsymbol{c}(x) = [c_1(x), c_2(x), \cdots, c_N(x)]$. The output $q(x)p(x)\boldsymbol{c}(x)$ of the convolutional encoder is sent to the AWGN channel using quadrature phase-shift keying (QPSK) modulation.

Because the CRC bits are added at the end of the sequence, the highest degree term of $f(x)$ is the bit that is first in time and first to enter the convolutional encoder. Consistent with this convention and in contrast to common representations, the highest degree terms of $\boldsymbol{c}(x)$ represent the most recent encoder input bits. Thus a convolution encoder with the generator $G(D) = [1 + D^3 + D^4, 1 + D + D^2]$ will have $\boldsymbol{c}(x) = [x^4 + x + 1, x^4 + x^3 + x^2]$.

The demodulated symbols are fed into a soft Viterbi decoder. The CRC decoder checks the n-bit sequence resulting from Viterbi decoding. An undetected error occurs when the receiver declares error-free decoding when the Viterbi decoder identified an incorrect codeword.

For a system employing a tail-biting convolutional code, it is discussed at the end of Section 4.2.1.3.

4.2 Undetected Error Probability Analysis

Let $e(x)$ be the polynomial of error bits in the Viterbi-decoded message so that the decoded n-bit sequence followed by ν zeros (for termination) is expressed as $q(x)p(x)x^{\nu} + e(x)$. If $e(x) \neq 0$ is divisible by $p(x)$, then this error is undetectable by the CRC decoder. This section presents two methods, the exclusion method and the construction method, to calculate the probability that a non-zero error $e(x)$ occurs that is undetectable.

This work focuses on the dominant error events of the convolutional code so that the analysis and the designed CRC codes are both most effective at high SNRs where overall behavior is well-characterized by these dominant events.

4.2.1 Exclusion Method

The exclusion method enumerates the possible error patterns of the convolutional code and excludes the patterns detectable by the CRC code. The probability of the unexcluded error patterns is the undetected error probability. The exclusion method filters out part of the distance spectrum of the convolutional code through a divisibility test to create the distance spectrum of the undetectable errors of the concatenated code.

4.2.1.1 Undetectable Single Error

An *error event* occurs when the decoded trellis path leaves the encoded trellis path once and rejoins it once. Let $e_{d,i}(x)$ be the i^{th} polynomial of message error bits that leads to a codeword distance d from the transmitted codeword. This error event has length $l_{d,i}$. Note that in this chapter, the term "distance" always refers to the convolutional code output Hamming distance.

If the received data frame contains only one error event, then the polynomial of message error bits can be expressed as $e(x) = x^g e_{d,i}(x)$, where $g \in [0, n+\nu-l_{d,i}]$. Note that the first (highest power) term of $e_{d,i}(x)$ is $x^{l_{d,i}-1}$ and the last (lowest power) term is x^ν because every error event starts with a one and ends with a one followed by ν consecutive zeros. If $e_{d,i}(x)$ is divisible by $p(x)$, this error event, including all of its offsets g, will be undetectable.

The probability of having such an error is bounded by

$$\mathrm{P}_{\mathrm{UD},1} \leq \sum_{d=d_{\mathrm{free}}}^{\infty} \sum_{i=1}^{a_d} \mathrm{I}(p(x) \,|\, e_{d,i}(x)) \max\{0, n+\nu-l_{d,i}+1\}\, \mathrm{P}(d)\,, \tag{4.1}$$

where d_{free} is the free distance of the convolutional code, a_d is the number of error events with output distance d, the indicator function $\mathrm{I}(\cdot)$ returns one when $e_{d,i}(x)$ is divisible by $p(x)$ and zero otherwise, and $\mathrm{P}(d)$ is the pairwise error probability of an error event with distance d. The subscript "1" in $\mathrm{P}_{\mathrm{UD},1}$ means that this probability only includes undetectable errors that are single error events. We call this type of error an *undetectable single error*.

For a QPSK system operated in an AWGN channel, $\mathrm{P}(d)$ can be computed using the tail probability function of standard normal distribution, i.e. Gaussian Q-function, as [VO09]

$$\mathrm{P}(d) = \mathrm{Q}\Big(\sqrt{2d\gamma}\Big) \leq \mathrm{Q}\Big(\sqrt{2d_{\mathrm{free}}\gamma}\Big)\, e^{-(d-d_{\mathrm{free}})\gamma}, \tag{4.2}$$

where $\gamma = E_{\mathrm{s}}/N_0$ is the signal-to-noise ratio (SNR) of a QPSK symbol, and E_{s} and $N_0/2$ denote the received symbol energy and one-dimensional noise variance, respectively. Note that the accuracy of (4.2) comes in part from a knowledge of d_{free}. A useful Q-function approximation when knowledge of d_{free} is not available is presented in [LD12].

To compute $\mathrm{P}_{\mathrm{UD},1}$ each error event $e_{d,i}$ must first identified as either divisible by $p(x)$ or not. One approach is to truncate (4.1) at $\tilde{d}$ to get an approximation, in which case all error events with distance $d \leq \tilde{d}$ can be stored and this set of error events can be tested for divisibility by the CRC polynomial $p(x)$. The choice of $\tilde{d}$ is based on the computational and storage capacity available to implement an efficient search such as [CJ89]. The required memory size to store the error events is proportional to $\sum_{d=d_{\mathrm{free}}}^{\tilde{d}} \sum_{i=1}^{a_d} l_{d,i}$.

The approximation of (4.1) can be quite tight if the probability of the terms with $d > \tilde{d}$ is negligible. However, assuming that all error events with $d > \tilde{d}$ are undetectable provides

$$\begin{aligned}\mathrm{P}_{\mathrm{UD},1} \leq & \sum_{d=\tilde{d}+1}^{\infty} n\, a_d\, \mathrm{P}(d) \\ & + \sum_{d=d_{\mathrm{free}}}^{\tilde{d}} \sum_{i=1}^{a_d} \{\mathrm{P}(d)\, \mathrm{I}(p(x) \mid e_{d,i}(x)) \max\{0, n+\nu-l_{d,i}+1\}\},\end{aligned} \tag{4.3}$$

where $l_{d,i}$ for $d > \tilde{d}$ is replaced with $\nu + 1$ because the shortest error event has length $\nu + 1$. Note that (4.3) can be computed because the error pattern $e_{d,i}$ is only required for $d \leq \tilde{d}$, and the distance spectrum a_d for $d > \tilde{d}$ provided by the transfer function [Vit71] can be used for the first term of (4.3) as in the frame error rate (FER) bounds of [ML99].

In addition to the undetectable single error events discussed above, an undetectable error could consist of two or more error events, even though each of the error events itself is detectable. We will first discuss the case with two error events, and then generalize it to multiple error events.

4.2.1.2 Undetectable Double Error

A double error involves two error events $e_{d_1,i_1}(x)$ and $e_{d_2,i_2}(x)$ with respective lengths l_{d_1,i_1} and l_{d_2,i_2}. To simplify notation, for $u \in \{1, 2\}$ let $e_u(x)$ and l_u refer to $e_{d_u,i_u}(x)$ and l_{d_u,i_u}, respectively. In a data frame with two error events, the polynomial of error bits in the message can be expressed as $e(x) = x^{g_1+g_2+l_2}e_1(x)+x^{g_2}e_2(x)$, where the exponents of two x's tell the locations of the two error events. Furthermore, $g_1 \geq 0$ represents the interval of symbols (gap) between two error events and satisfies $g_1 + g_2 + l_1 + l_2 \leq n + \nu$. If $x^{g_1+l_2}e_1(x) + e_2(x)$ is divisible by $p(x)$, the error is an *undetectable double error*. Its length is $l_1 + l_2 + g_1$ and its offset is g_2.

An upper bound of the probability of an undetectable double error occurring in the codeword is given by

$$\begin{aligned} \mathrm{P}_{\mathrm{UD},2} \leq & \sum_{d_1=d_{\mathrm{free}}}^{\infty} \sum_{d_2=d_{\mathrm{free}}}^{\infty} \sum_{i_1=1}^{a_{d_1}} \sum_{i_2=1}^{a_{d_2}} \sum_{g_1=0}^{n+\nu-l_1-l_2} \mathrm{P}(d_1+d_2) \\ & \cdot \mathrm{I}\big(p(x) \mid x^{g_1+l_2}e_1(x)+e_2(x)\big)\,(n+\nu-l_1-l_2-g_1+1)\,. \end{aligned} \tag{4.4}$$

The distance of the double error event is simply the sum of the individual distances because the error events are completely separated as shown in Fig. 4.2. Two error events that overlap are simply a longer single error event, which was treated in Section 4.2.1.1.

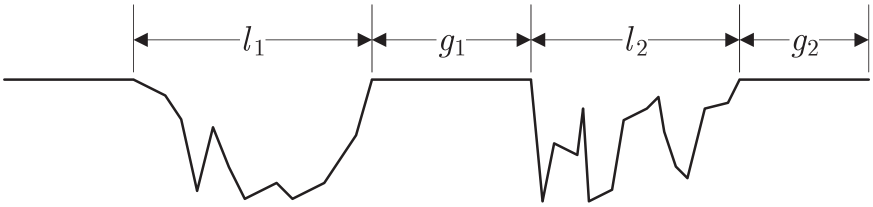


Figure 4.2: An illustration of two error events.

Computation of (4.4) exactly is problematic because it requires infinite search depth. Replacing the l_1 and l_2 of large-distance terms in (4.4) with $\nu+1$ yields a more computation-friendly upper bound similar to (4.3) as follows:

$$\begin{aligned} \mathrm{P}_{\mathrm{UD},2} \leq & \sum_{(d_1,d_2)\in\mathbf{D}_{\tilde{d},2}} \sum_{i_1=1}^{a_{d_1}} \sum_{i_2=1}^{a_{d_2}} \sum_{g_1=0}^{n+\nu-l_1-l_2} \mathrm{P}(d_1+d_2)\,\mathrm{I}\big(p(x) \mid x^{g_1+l_2}e_1(x)+e_2(x)\big) \\ & \cdot (n+\nu-l_1-l_2-g_1+1) \\ & + \sum_{(d_1,d_2)\notin\mathbf{D}_{\tilde{d},2}} \frac{1}{2}\,(n-\nu)\,(n-\nu-1)a_{d_1}a_{d_2}\,\mathrm{P}(d_1+d_2)\,, \end{aligned} \tag{4.5}$$

where $\mathbf{D}_{\tilde{d},2} = \Big\{(d_1,d_2)\Big|d_1,d_2 \geq d_{\mathrm{free}}, d_1+d_2 \leq \tilde{d}\Big\}$.

Because computational complexity limits the *single* error event distance we can search, it is feasible to replace $\tilde{d}$ in $\mathbf{D}_{\tilde{d},2}$ with $\tilde{d}+d_{\mathrm{free}}$. This is not particularly helpful because we have already assumed errors with distance $d > \tilde{d}$ are negligible or undetectable during the calculation of undetectable single errors. We can also replace all l_1 and l_2 in (4.5) with $\nu+1$ and provide another upper bound which does not require any length information.

As described in Appendix 4.A, the number of g_1 values at which to check the divisibility of $x^{g_1+l_2}e_1(x)+e_2(x)$ by $p(x)$ can be significantly reduced from $n+\nu-l_1-l_2+1$.

4.2.1.3 Total Undetected Error Probability

In general, s error events could possibly form an undetectable s-tuple error, whether each of them is detectable or not. The error bits in the message can be expressed as

$$e(x) = \sum_{u=1}^{s} \left(\prod_{v=u+1}^{s} x^{g_v + l_v} \right) x^{g_u} e_u(x). \tag{4.6}$$

This combined error will be undetectable if $e(x)$ is divisible by $p(x)$. Therefore, the probability of undetectable s-tuple errors $\mathrm{P}_{\mathrm{UD},s}$ can be approximated or bounded in the same way as (4.5). For simpler notation, define sets

$$\begin{aligned}
\mathbf{D}_{\tilde{d},s} &= \left\{ (d_1, \cdots, d_s) \,\middle|\, d_u \geq d_{\text{free}} \ \ \forall u, \sum_{u=1}^{s} d_u \leq \tilde{d} \right\} \\
\mathbf{I}_s &= \left\{ (i_1, \cdots, i_s) \,|\, i_u \in [1, a_{d_u}] \ \ \forall u \in [1, s] \right\} \\
\mathbf{G}_s &= \left\{ (g_1, \cdots, g_s) \,\middle|\, g_u \geq 0 \ \ \forall u, \sum_{u=1}^{s} g_u \leq n + \nu - \sum_{u=1}^{s+1} l_u \right\}.
\end{aligned}$$

Note that $\mathbf{G}_s$ is determined by all d_u and i_u, and $\mathbf{I}_s$ is determined by all d_u.

Since an undetectable error may consist of any number of error events, the probability of having an undetectable error in the codeword P_{UD} is upper bounded by $\sum_{s=1}^{\infty} \mathrm{P}_{\mathrm{UD},s}$. Using the computation-friendly bound of each term such as (4.3) and (4.5), we obtain

$$\begin{aligned}
\mathrm{P}_{\mathrm{UD}} \leq \sum_{s=1}^{\infty} \mathrm{P}_{>\tilde{d},s} &+ \sum_{d_1 \in \mathbf{D}_{\tilde{d},1}} \left\{ \mathrm{P}(d_1) \sum_{i_1 \in \mathbf{I}_1} \mathrm{I}(p(x) \mid e_1(x)) \max\left\{0, n + \nu - l_1 + 1\right\} \right\} \\
&+ \sum_{s=2}^{\infty} \left\{ \sum_{(d_1, \cdots, d_s) \in \mathbf{D}_{\tilde{d},s}} \ \sum_{(i_1, \cdots, i_s) \in \mathbf{I}_s} \ \sum_{(g_1, \cdots, g_{s-1}) \in \mathbf{G}_{s-1}} \mathrm{I}(p(x) \mid e(x)) \right. \\
&\cdot \left. \left(n + \nu - \sum_{u=1}^{s} l_u - \sum_{u=1}^{s-1} g_u + 1 \right) \mathrm{P}\left(\sum_{u=1}^{s} d_u \right) \right\},
\end{aligned} \tag{4.7}$$

where the composition of $e(x)$ depends on the number of error events s as in (4.6) and $\mathrm{P}_{>\tilde{d},s}$ is the sum of probability of all s-tuple errors whose distances are greater than $\tilde{d}$.

The probability sum of all large-distance s-tuple errors is

$$\mathrm{P}_{>\tilde{d},s} = \sum_{(d_1, \cdots, d_s) \notin \mathbf{D}_{\tilde{d},s}} \binom{n + \nu - s\nu}{s} \left(\prod_{u=1}^{s} a_{d_u} \right) \mathrm{P}\left(\sum_{u=1}^{s} d_u \right), \tag{4.8}$$

where the combinatorial number represents the number of ways to have s length-$(\nu+1)$ error events in a length-$(n+\nu)$ sequence. Using (4.2) $\mathrm{P}_{>\tilde{d},1}$ can be upper bounded by

$$\mathrm{P}_{>\tilde{d},1} \leq \mathrm{Q}\Big(\sqrt{2d_{\text{free}}\gamma}\Big)\, e^{d_{\text{free}}\gamma} \left\{ \bar{\mathrm{P}} - \sum_{d_1 \in \mathbf{D}_{\tilde{d},1}} n\, a_{d_1} e^{-d_1\gamma} \right\}, \tag{4.9}$$

where $\bar{\mathrm{P}}$ is defined as $\bar{\mathrm{P}} \triangleq n\mathrm{T}(D,L)|_{D=e^{-\gamma},L=1}$ using the transfer function [Vit71]

$$\mathrm{T}(D,L) = \sum_{d=d_{\text{free}}}^{\infty} \sum_{i=1}^{a_d} D^d L^{l_{d,i}}, \tag{4.10}$$

where the exponents of D and L indicate the output Hamming distances and lengths of the error events, respectively. Therefore, the sum of $\mathrm{P}_{>\tilde{d},s}$ terms can be upper bounded by

$$\begin{aligned}
\sum_{s=1}^{\infty} \mathrm{P}_{>\tilde{d},s} &\leq \sum_{s=1}^{\infty} \sum_{(d_1,\cdots,d_s)\notin \mathbf{D}_{\tilde{d},s}} \frac{n^s}{s!} \left(\prod_{u=1}^{s} a_{d_u}\right) \mathrm{P}\left(\sum_{u=1}^{s} d_u\right) \\
&\leq \mathrm{Q}\Big(\sqrt{2d_{\text{free}}\gamma}\Big)\, e^{d_{\text{free}}\gamma} \sum_{s=1}^{\infty} \left\{ \frac{n^s}{s!} \sum_{(d_1,\cdots,d_s)\notin \mathbf{D}_{\tilde{d},s}} \left(\prod_{u=1}^{s} a_{d_u} e^{-d_u\gamma}\right)\right\} && \text{(4.10a)} \\
&= \mathrm{Q}\Big(\sqrt{2d_{\text{free}}\gamma}\Big)\, e^{d_{\text{free}}\gamma} \sum_{s=1}^{\infty} \left\{ \frac{1}{s!}\bar{\mathrm{P}}^s - \frac{n^s}{s!} \sum_{(d_1,\cdots,d_s)\in \mathbf{D}_{\tilde{d},s}} \left(\prod_{u=1}^{s} a_{d_u} e^{-d_u\gamma}\right)\right\} && \text{(4.10b)} \\
&= \mathrm{Q}\Big(\sqrt{2d_{\text{free}}\gamma}\Big)\, e^{d_{\text{free}}\gamma} \left\{ e^{\bar{\mathrm{P}}} - 1 - \sum_{s=1}^{\infty} \frac{n^s}{s!} \sum_{(d_1,\cdots,d_s)\in \mathbf{D}_{\tilde{d},s}} \left(\prod_{u=1}^{s} a_{d_u} e^{-d_u\gamma}\right)\right\}. && \text{(4.10c)}
\end{aligned}$$

The bound of Gaussian Q-function (4.2) is used in (4.10a), and the transfer function is used to evaluate the sum of all s-tuple errors in (4.10b). Using (4.7) and (4.10c), a bound of P_{UD} can be calculated. In fact, when an undetectable error occurs in a codeword, the receiver may still detect an error if a detectable error happens somewhere else in the codeword. Therefore, P_{UD}, which is the probability of having an undetectable error in the codeword, is an upper bound of the undetected error probability. When channel error rate is low, having a detected error occur along with the undetected error is a rare event so that our bound will be tight.

If we were to consider a tail-biting convolutional code, only a minor modification is required. Although there are no terminating bits, the number of locations to start an error event is still n. Moreover, the number of available offsets of an undetectable error is n, which is independent of its length. Therefore, to obtain a bound for the undetected error probability, we just need to let the lengths of all undetectable errors be one and $\nu = 0$, and the derivation from (4.7) to (4.10c) will be valid.

Due to the limitation of searchable depth $\tilde{d}$ of error events, the exclusion method is only useful when undetectable errors with distances $d \leq \tilde{d}$ dominate. However, this requirement could be violated by a powerful CRC code that detects the short-distance errors. The next subsection introduces the construction method, which allows the search depth to increase to distance $\hat{d} > \tilde{d}$.

4.2.2 Construction Method

The construction method utilizes the fact that all undetectable errors at the CRC decoder input are multiples of the CRC generator $p(x)$. This method constructs an equivalent convolutional encoder $\boldsymbol{c_{eq}}(x) = p(x)\boldsymbol{c}(x)$ to isolate these errors. The set of non-zero codewords of $\boldsymbol{c_{eq}}(x)$ is exactly the set of erroneous codewords (given an all-zeros transmission) that lead to undetectable errors for the concatenation of the CRC generator polynomial and the original convolutional encoder. Thus the probability of undetectable error is exactly the FER of $\boldsymbol{c_{eq}}(x)$.

Fig. 4.3 shows an example of how $\boldsymbol{c_{eq}}(x)$ is constructed from $\boldsymbol{c}(x)$ and $p(x)$, where $q'(x)$ with $m+\nu$ trailing zeros is the input that generates the undetectable erroneous codeword. The input/output behavior of $\boldsymbol{c_{eq}}(x)$ is exactly the same as the concatenation of $p(x)$ and $\boldsymbol{c}(x)$. For $p(x)$ with m memory elements and $\boldsymbol{c}(x)$ with ν memory elements, $\boldsymbol{c_{eq}}(x)$ has $m+\nu$ memory elements. At a given time, the state of the original encoder $\boldsymbol{c}(x)$ can be inferred from the state of $\boldsymbol{c_{eq}}(x)$ because

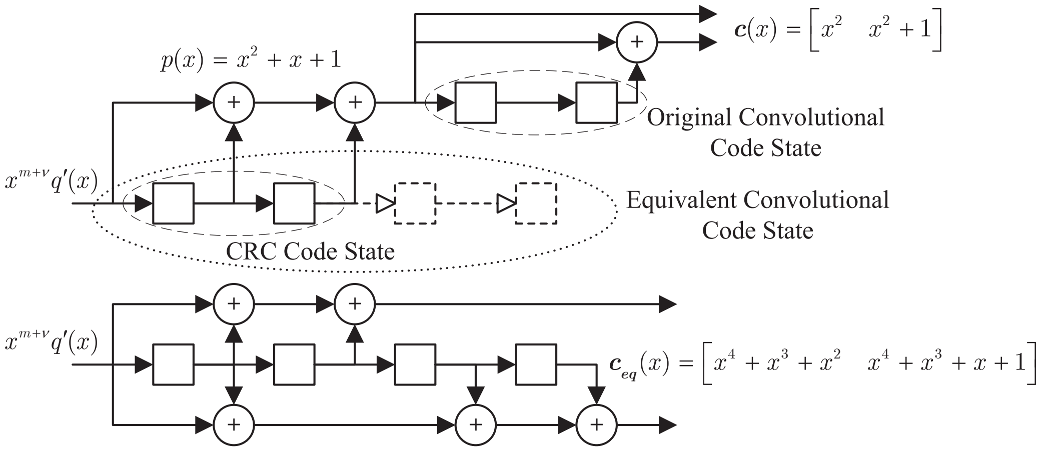

$p(x) = x^2 + x + 1$
$\boldsymbol{c}(x) = \left[x^2 \quad x^2+1\right]$
$x^{m+\nu}q'(x)$
Original Convolutional Code State
Equivalent Convolutional Code State
CRC Code State
$x^{m+\nu}q'(x)$
$\boldsymbol{c}_{eq}(x) = \left[x^4+x^3+x^2 \quad x^4+x^3+x+1\right]$

Figure 4.3: An example of CRC code, original convolutional code, and equivalent convolutional code.

the state of $\boldsymbol{c_{eq}}(x)$ contains the last $m+\nu$ inputs to $p(x)$ which are sufficient to compute the last ν outputs of $p(x)$, which exactly comprise the state of $\boldsymbol{c}(x)$.

4.2.2.1 States of the Equivalent Encoder

Define the all-zero state $\mathbf{S}^{\mathrm{Z}}$ to be the state where all memory elements of the equivalent encoder are zero. When the equivalent encoder is in $\mathbf{S}^{\mathrm{Z}}$, then the original encoder is also in its zero state. Avoiding trivial cases by assuming that the x^{m-1} and x^0 coefficients of $p(x)$ are 1, there are 2^m-1 equivalent encoder states in addition to $\mathbf{S}^{\mathrm{Z}}$ that correspond to the all-zero state of the original encoder. To see this, consider the top diagram in Fig. 4.3 in which the equivalent encoder state is shown as the state of the CRC encoder extended with ν additional memories. Note that whatever state the equivalent encoder is in, there is a sequence of ν bits that will produce ν zeros at the output of the CRC encoder that will drive the original encoder state to zero. The specific set of ν bits is a function of the m-bits of state in the CRC encoder. Thus for any m-bits pattern there is a corresponding $(m+\nu)$-bit state of the equivalent encoder that corresponds to the original encoder being in the zero state.

Table 4.1: An example of state types with $p(x) = x^2 + x + 1$ and $\nu = 2$.

	State	Equivalent Code		Original Code	
Time	Type	State	Input	State	Input
0	$\mathbf{S}^{\mathrm{Z}}$	0000	1	00	1
1	$\mathbf{S}^{\mathrm{N}}$	0001	1	01	0
2	$\mathbf{S}^{\mathrm{N}}$	0011	0	10	0
3	$\mathbf{S}^{\mathrm{D}}$	0110	1	00	0
4	$\mathbf{S}^{\mathrm{D}}$	1101	0	00	1
5	$\mathbf{S}^{\mathrm{N}}$	1010	0	01	1
6	$\mathbf{S}^{\mathrm{N}}$	0100	0	11	0
7	$\mathbf{S}^{\mathrm{N}}$	1000	0	10	0
8	$\mathbf{S}^{\mathrm{Z}}$	0000		00	

We call the $2^m - 1$ non-zero equivalent encoder states for which the original encoder state is zero *detectable-zero states*, forming a set $\mathbf{S}^{\mathrm{D}}$, because they correspond to the termination of a detectable error event in the original encoder. The remaining $2^{m+\nu} - 2^m$ states of the equivalent code are called *non-zero states*, forming a set $\mathbf{S}^{\mathrm{N}}$, because the corresponding states of the original code are not zero. To terminate an error event in the original encoder, the trellis of the equivalent code transitions from a state in $\mathbf{S}^{\mathrm{N}}$ to $\mathbf{S}^{\mathrm{Z}}$ or to a state in $\mathbf{S}^{\mathrm{D}}$. If the transition is to $\mathbf{S}^{\mathrm{D}}$ the cumulative errors are detectable because the portion of $q'(x)p(x)$ till this moment is not divisible by $p(x)$. If the transition is to $\mathbf{S}^{\mathrm{D}}$, the cumulative errors are detectable because the portion of $q'(x)p(x)$ till this moment is divisible by $p(x)$.

Table 4.1 shows an example using the set-up shown in Fig. 4.3 with $p(x) = x^2 + x + 1$, $q'(x) = x^3 + x^2 + 1$, and $\nu = 2$. The original encoder $\boldsymbol{c}(x)$ does not need to be specified for the results in Table 4.1 because any feedforward encoder will produce the same state sequence. The zero state of the original code is visited at time 0, 3, 4, and 8. At time 0, the state begins from $\mathbf{S}^{\mathrm{Z}}$. At time 3, the first $\boldsymbol{c}(x)$ error event ends. This error event is detectable if the codeword ends at time 3 because the input to the original code, i.e. x^2, is not divisible by $p(x)$. The state remains in $\mathbf{S}^{\mathrm{D}}$ at time 4. At time 5 a second $\boldsymbol{c}(x)$ error event begins. At time 8, the $\boldsymbol{c_{eq}}(x)$ state returns to $\mathbf{S}^{\mathrm{Z}}$, which is not only the end of the second $\boldsymbol{c}(x)$ error event but also the end of an undetectable double error.

Note that $\boldsymbol{c_{eq}}(x)$ is catastrophic because its generator polynomials have a common factor $p(x)$. The catastrophic behavior is expressed through the zero distance loops that occur as the equivalent encoder traverses a sequence of states in $\mathbf{S}^{\mathrm{D}}$ while the original convolutional encoder stays in the zero state. Thus time spent in $\mathbf{S}^{\mathrm{D}}$ between $\boldsymbol{c}(x)$ error events can lengthen an undetectable error without increasing its distance.

4.2.2.2 Error Events in the Equivalent Encoder

Since the all-zero codeword is assumed to be sent, the correct path remains in $\mathbf{S}^{\mathrm{Z}}$ forever. An error event in the equivalent encoder occurs if the trellis path leaves $\mathbf{S}^{\mathrm{Z}}$ and returns $\mathbf{S}^{\mathrm{Z}}$ without any visits to $\mathbf{S}^{\mathrm{Z}}$ in between. We will classify these error events according to the number of times it enters $\mathbf{S}^{\mathrm{D}}$ from $\mathbf{S}^{\mathrm{N}}$ during the deviation. If a trellis path enters $\mathbf{S}^{\mathrm{D}}$ from $\mathbf{S}^{\mathrm{N}}$ $s-1$ times, then it is classified as an undetectable s-tuple error. In an undetectable s-tuple error, there are exactly s segments of consecutive transitions between states in $\mathbf{S}^{\mathrm{N}}$, which correspond to s error events of the original encoder. These segments are separated by segments of consecutive transitions between states in $\mathbf{S}^{\mathrm{D}}$.

Fig. 4.4 illustrates an undetectable triple error in a system with $\nu = 2$ and $m = 2$. The three error events of the original code are separated by visits to $\mathbf{S}^{\mathrm{D}}$. The path can leave $\mathbf{S}^{\mathrm{D}}$ right after entering it as shown between the first and the second error events; the path can also stay in $\mathbf{S}^{\mathrm{D}}$ for a while and then leave $\mathbf{S}^{\mathrm{D}}$

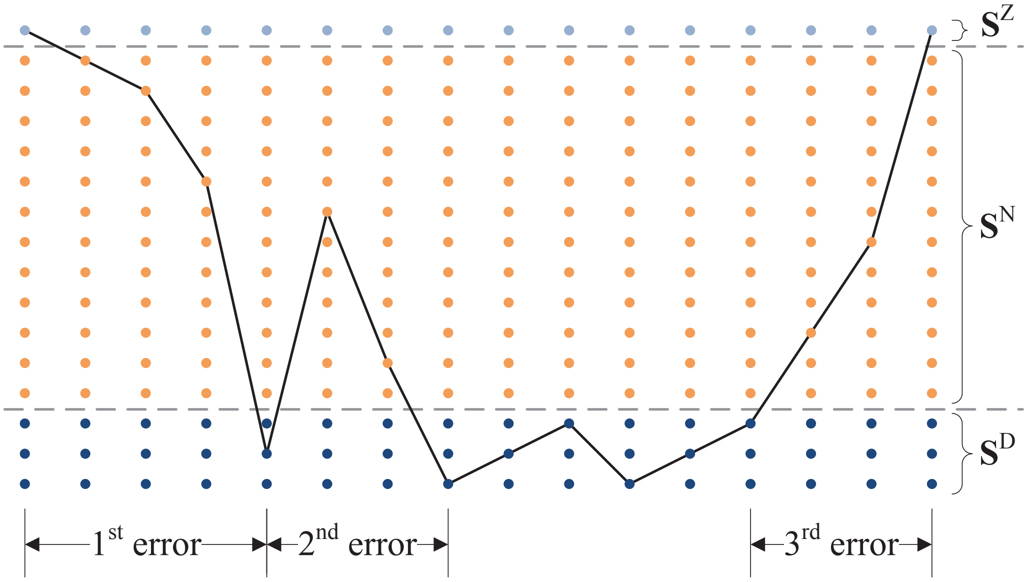

$\mathbf{S}^{\mathrm{Z}}$
$\mathbf{S}^{\mathrm{N}}$
$\mathbf{S}^{\mathrm{D}}$
1st error
2nd error
3rd error

Figure 4.4: A trellis diagram of an equivalent convolutional code with $\nu = 2$ and $m = 2$ having an undetectable triple error. The states are reordered such that the all-zero state is at the top and the detectable-zero states are at the bottom.

from a state different from where it enters $\mathbf{S}^{\mathrm{D}}$ as shown between the second and the third error events. Note that $\mathbf{S}^{\mathrm{Z}}$ is never directly connected to any state of $\mathbf{S}^{\mathrm{D}}$.

The probability of undetectable single errors is bounded in a similar way as (4.3) by

$$\mathrm{P}_{\mathrm{UD},1} \leq \mathrm{P}_{>\hat{d},1} + \sum_{d \in \mathbf{D}_{\hat{d},1}} \sum_{i=1}^{a_d^{\mathrm{ZZ}}} \max\left\{0, n + \nu - l_{d,i}^{\mathrm{ZZ}} + 1\right\} \mathrm{P}(d), \tag{4.11}$$

where a_d^{ZZ} is the number of error events with distance d starting from $\mathbf{S}^{\mathrm{Z}}$ and ending at $\mathbf{S}^{\mathrm{Z}}$ while never traversing a state in $\mathbf{S}^{\mathrm{D}}$. The length $l_{d,i}^{\mathrm{ZZ}}$ is the length of the i^{th} error event counted in a_d^{ZZ}. Note that this expression requires the distance spectrum a_d^{ZZ} and length spectrum $l_{d,i}^{\mathrm{ZZ}}$ for $d \in \left[d_{\mathrm{free}}, \hat{d}\right]$ obtained using computer search.

Unlike (4.3), (4.11) does not need to check the divisibility of error events. Hence, there is no need to store the error patterns anymore but only their distances and lengths. Consequently the search depth $\hat{d}$ can go beyond $\tilde{d}$.

In fact, it is possible to release the search depth limitation $\hat{d}$ and calculate a_d^{ZZ} and $l_{d,i}^{\mathrm{ZZ}}$ through an altered transfer function. Since the trellis path is not allowed to enter $\mathbf{S}^{\mathrm{D}}$, the altered transfer function can be computed using a modified transition matrix whose rows and columns of $\mathbf{S}^{\mathrm{D}}$ are deleted. However, there are $2^{m+\nu}-2^m+1$ remaining states after the deletion and the size of the modified transition matrix could still be huge. Therefore, this alternative method is only applicable to the cases with small $m+\nu$.

The probability of an undetectable double error is determined by the distances of its two error events, where the first leaves $\mathbf{S}^{\mathrm{Z}}$ and enters a state in $\mathbf{S}^{\mathrm{D}}$ while the second leaves a state in $\mathbf{S}^{\mathrm{D}}$ and enters $\mathbf{S}^{\mathrm{Z}}$. The length of this undetectable double error depends on not only the lengths of the two error events but also the spacing between them, i.e. the number of intervening state transitions within $\mathbf{S}^{\mathrm{D}}$. In the example shown in Fig. 4.4, we observe that the spacing between error events could be any non-negative integer as long as the double error does not exceed the codeword length.

Since states in $\mathbf{S}^{\mathrm{D}}$ correspond to the all-zero state of the original code, one can prove that each of them can lead to a state in $\mathbf{S}^{\mathrm{D}}$ with a proper input bit which generates a zero at the original convolutional encoder input. For example, consider a degree-2 primitive polynomial $p(x) = x^2 + x + 1$ and $\nu = 2$. We have $\mathbf{S}^{\mathrm{D}} = \{(0110), (1101), (1011)\}$, and the states in $\mathbf{S}^{\mathrm{D}}$ lead to one another. However, for certain $p(x)$, it may not be able to traverse all states in $\mathbf{S}^{\mathrm{D}}$ without leaving $\mathbf{S}^{\mathrm{D}}$. For example, consider a degree-2 non-primitive polynomial $p(x) = x^2 + 1$ and $\nu = 2$. We have $\mathbf{S}^{\mathrm{D}} = \{(0101), (1010), (1111)\}$. The first two states lead to each other but the third state only leads to itself if the transition is not allowed to leave $\mathbf{S}^{\mathrm{D}}$.

Let $\mathrm{S}_i^{\mathrm{D}}$ be the i^{th} state in $\mathbf{S}^{\mathrm{D}}$, where $i \in [1, 2^m - 1]$. Define $\boldsymbol{\Delta}_i$ as the subset of $\mathbf{S}^{\mathrm{D}}$ composed of all states connected with $\mathrm{S}_i^{\mathrm{D}}$ with all intermediate states in

$\mathbf{S}^{\mathrm{D}}$. Let $\delta_{i,j} \leq |\boldsymbol{\Delta}_i|$ be the number of hops required to go from $\mathrm{S}_i^{\mathrm{D}}$ to $\mathrm{S}_j^{\mathrm{D}}$ without leaving $\mathbf{S}^{\mathrm{D}}$ for $i \in [1, 2^m - 1]$ and $\mathrm{S}_j^{\mathrm{D}} \in \boldsymbol{\Delta}_i$, where $\delta_{i,j} = 0$ if $i = j$. The sets $\boldsymbol{\Delta}_i$ and $\boldsymbol{\Delta}_j$ are identical if $\mathrm{S}_j^{\mathrm{D}} \in \boldsymbol{\Delta}_i$, and the distinct $\boldsymbol{\Delta}_i$'s form a partition of $\mathbf{S}^{\mathrm{D}}$. Appendix 4.B shows that $\boldsymbol{\Delta}_i$ and the x-cyclotomic coset modulo $p(x)$ discussed in Appendix 4.A are equivalent. Thus, all sets $\boldsymbol{\Delta}_i$ are identical if $p(x)$ is a primitive polynomial.

Similar to (4.5), the probability of undetectable double errors can be bounded by

$$
\begin{aligned}
\mathrm{P}_{\mathrm{UD},2} \leq \mathrm{P}_{>\hat{d},2} + & \sum_{(d_1,d_2)\in\mathbf{D}_{\hat{d},2}} \mathrm{P}(d_1+d_2) \\
& \cdot \sum_{\phi=1}^{2^m-1} \sum_{\mathrm{S}_\psi^{\mathrm{D}}\in\boldsymbol{\Delta}_\phi} \sum_{i_1=1}^{a_{d_1}^{\mathrm{ZD},\phi}} \sum_{i_2=1}^{a_{d_2}^{\mathrm{DZ},\psi}} \sum_{t_1=0}^{\left\lfloor \frac{n+r-l^{\min}}{|\boldsymbol{\Delta}_\phi|} \right\rfloor} \big(n+\nu-l^{\min}-|\boldsymbol{\Delta}_\phi|\, t_1+1\big), \qquad (4.12)
\end{aligned}
$$

where

$$
l^{\min} = l_{d_1,i_1}^{\mathrm{ZD},\phi} + l_{d_2,i_2}^{\mathrm{DZ},\psi} + \delta_{\phi,\psi} \qquad (4.13)
$$

is the shortest possible length of the undetectable double error specified by the combination of indices $(d_1, d_2, \phi, \psi, i_1, i_2)$.

In (4.12), ϕ is the index of the state at which the trellis enters $\mathbf{S}^{\mathrm{D}}$ at the end of the first error event, and ψ is the index of the state at which the trellis leaves $\mathbf{S}^{\mathrm{D}}$ at the beginning of the second error event. The number of error events starting at $\mathbf{S}^{\mathrm{Z}}$ and ending at $\mathrm{S}_\phi^{\mathrm{D}}$ with distance d_1 is $a_{d_1}^{\mathrm{ZD},\phi}$, and the $i_1{}^{\mathrm{th}}$ error event of them has length $l_{d_1,i_1}^{\mathrm{ZD},\phi}$, where both numbers are obtained by computer search. The variables $a_{d_2}^{\mathrm{DZ},\psi}$ and $l_{d_2,i_2}^{\mathrm{DZ},\psi}$ are defined in a similar way while the error event starts in $\mathrm{S}_\psi^{\mathrm{D}}$ and ends in $\mathbf{S}^{\mathrm{Z}}$. Furthermore, t_1 specifies the number of cycles the trellis stays in $\boldsymbol{\Delta}_\phi$ and its upper limit makes sure that the total length of the undetectable double error does not exceed $n+\nu$, the number of trellis stages in the codeword. As in (4.5) $\mathrm{P}_{>\hat{d},2}$ can often be neglected because terms with $d_1 + d_2 > \hat{d}$ have negligible probability.

Although the number and lengths of error events connecting $\mathbf{S}^{\mathrm{Z}}$ and states in $\mathbf{S}^{\mathrm{D}}$ are obtained by computer searches, the required search depth is only $\hat{d} - d_{\text{free}}$. Moreover, $a_{d_1}^{\mathrm{ZD},\phi}$ and $l_{d_1,i_1}^{\mathrm{ZD},\phi}$ for all ϕ can be found while searching for a_d^{ZZ} and $l_{d,i}^{\mathrm{ZZ}}$ because these two types of error events both start from $\mathbf{S}^{\mathrm{Z}}$. Regarding $a_{d_2}^{\mathrm{DZ},\psi}$ and $l_{d_2,i_2}^{\mathrm{DZ},\psi}$, the associated error events start from $2^m - 1$ different states and can be found through $2^m - 1$ separated searches. Nevertheless, since these error events end at $\mathbf{S}^{\mathrm{Z}}$, only one backward search is necessary to capture all of them. In the backward search, bits in shift registers move backward. One can simply treat x as x^{-1} in polynomial representations and apply the same search algorithm.

An easier way to search for error events ending at $\mathbf{S}^{\mathrm{Z}}$ is to utilize the search result of error events starting from $\mathbf{S}^{\mathrm{Z}}$. See Appendix 4.C for detail.

4.2.2.3 Undetected Error Probability

An undetectable s-tuple error is composed of several parts: one error event from $\mathbf{S}^{\mathrm{Z}}$ to a state in $\mathbf{S}^{\mathrm{D}}$, $s-1$ paths inside $\mathbf{S}^{\mathrm{D}}$, $s-2$ error events from a state in $\mathbf{S}^{\mathrm{D}}$ to a state in $\mathbf{S}^{\mathrm{D}}$ with visits to $\mathbf{S}^{\mathrm{N}}$ in between, and one error event from a state in $\mathbf{S}^{\mathrm{D}}$ to $\mathbf{S}^{\mathrm{Z}}$. Let ϕ_u and ψ_u be the indices of the start and end states of the u^{th} transitions in $\mathbf{S}^{\mathrm{D}}$, respectively. Also, let the number of error events started at $\mathrm{S}_{\psi_{u-1}}^{\mathrm{D}}$ and ended at $\mathrm{S}_{\phi_u}^{\mathrm{D}}$ with distance d_u be $a_{d_u}^{\mathrm{DD},\psi_{u-1},\phi_u}$ and the length of the $i_u{}^{\text{th}}$ error event of them be $l_{d_u,i_u}^{\mathrm{DD},\psi_{u-1},\phi_u}$, where both numbers are obtained by computer search. Although we need to perform $2^m - 1$ separated searches to obtain all $a_{d_u}^{\mathrm{DD},\psi_{u-1},\phi_u}$ and $l_{d_u,i_u}^{\mathrm{DD},\psi_{u-1},\phi_u}$, a search depth of $\hat{d} - (s-1)\, d_{\text{free}}$ is sufficient. Similar to the search for error events ending at $\mathbf{S}^{\mathrm{Z}}$, there is an easier approach presented in Appendix 4.C.

To simplify the notation, define the following sets:

$$\mathbf{\Phi}_s = \{(\phi_1, \cdots, \phi_s) | \phi_u \in [1, 2^m - 1] \ \ \forall u \in [1, s]\}$$

$$\mathbf{\Psi}_s = \{(\psi_1, \cdots, \psi_s) | \psi_u \in \mathbf{\Delta}_{\phi_u} \ \ \forall u \in [1, s]\}$$

$$\begin{aligned}\mathbf{I}'_s = \Big\{(i_1, \cdots, i_s)\Big| i_1 \in \Big[1, a_{d_1}^{\mathrm{ZD},\phi_1}\Big], i_s \in \Big[1, a_{d_s}^{\mathrm{DZ},\psi_{s-1}}\Big],\\ i_u \in \Big[1, a_{d_u}^{\mathrm{DD},\psi_{u-1},\phi_u}\Big]\ \forall u \in [2, s-1]\Big\}\\ \mathbf{T}_s = \left\{(t_1, \cdots, t_s)\middle| t_u \geq 0\ \ \forall u \in [1, s], \sum_{u=1}^{s} |\mathbf{\Delta}_{\phi_u}| t_u \leq n + r - l_{s+1}^{\min}\right\},\end{aligned}$$

where $l_s^{\min}$ is the shortest possible length of the undetectable s-tuple error specified in a similar way as (4.13) and given by

$$l_s^{\min} = l_{d_1,i_1}^{\mathrm{ZD},\phi_1} + \sum_{u=2}^{s-1} l_{d_u,i_u}^{\mathrm{DD},\psi_{u-1},\phi_u} + l_{d_s,i_s}^{\mathrm{DZ},\psi_{s-1}} + \sum_{u=1}^{s-1} \delta_{\phi_u,\psi_u}. \tag{4.14}$$

Similar to (4.7), the probability of having an undetectable error is now bounded by

$$\begin{aligned}\mathrm{P}_{\mathrm{UD}} \leq \sum_{s=1}^{\infty} \mathrm{P}_{>\hat{d},s} + \sum_{d_1 \in \mathbf{D}_{\hat{d},1}} \sum_{i=1}^{a_{d_1}^{\mathrm{ZZ}}} \max\big\{0, n+\nu-l_{d_1,i}^{\mathrm{ZZ}}+1\big\}\mathrm{P}(d_1)\\ +\sum_{s=2}^{\infty} \sum_{(d_1,\cdots,d_s)\in\mathbf{D}_{\hat{d},s}} \mathrm{P}\left(\sum_{u=1}^{s} d_u\right) \sum_{(\phi_1,\cdots,\phi_{s-1})\in\mathbf{\Phi}_{s-1}} \sum_{(\psi_1,\cdots,\psi_{s-1})\in\mathbf{\Psi}_{s-1}}\\ \sum_{(i_1,\cdots,i_s)\in\mathbf{I}'_s} \sum_{(t_1,\cdots,t_{s-1})\in\mathbf{T}_{s-1}} \left(n+\nu-l_s^{\min}-\sum_{u=1}^{s-1} |\mathbf{\Delta}_{\phi_u}| t_u+1\right).\end{aligned} \tag{4.15}$$

The first term is the probability sum of all large-distance errors, which are assumed to be undetectable, and can be calculated using (4.10c). By letting $\mathrm{P}_{>\hat{d},s} = 0$, we obtain an approximation. By letting all l^{ZZ}, l^{ZD}, l^{DD}, and l^{DZ} equal to $\nu + 1$, we obtain a looser bound which does not require any length information. These two techniques are applicable to every $\mathrm{P}_{\mathrm{UD},s}$, including (4.11) and (4.12).

The main benefit of the construction method is that it is often able to search deeper than the exclusion method because the output pattern is not required. However, the required memory size scales with the number of states $2^{m+\nu}$ rather than 2^{ν} so this approach can encounter difficulty in analyzing high-order CRC codes. In contrast, the error events searched in the exclusion method belong to the original convolutional code, whose number of states is just 2^{ν} and is independent of the degree of the CRC code. As explained in Section 4.3, we found it useful

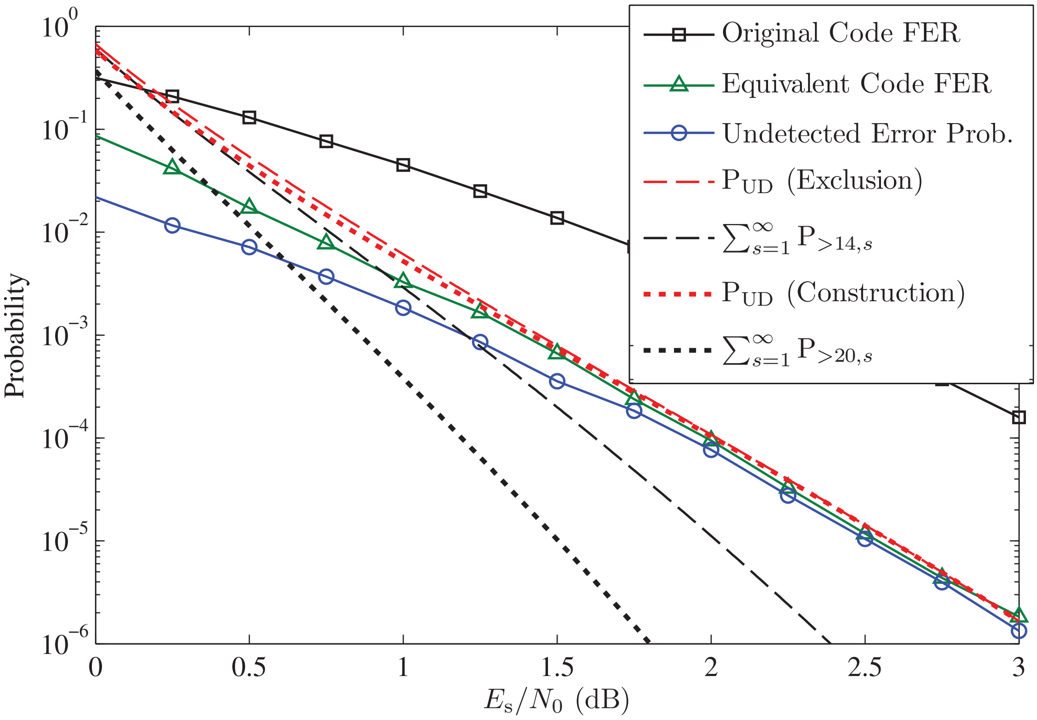


Figure 4.5: A comparison of the simulated undetected error probability with the simulated frame error rate of the equivalent code and the analyses from the exclusion and construction methods. The system is equipped with the CRC code $p(x) = x^3+x+1$ and the convolutional code $(23, 35)_8$. The CRC codeword length is $n = 1024$ bits.

to draw on both approaches as we searched for optimal CRC polynomials for a specific convolutional code.

Fig. 4.5 compares the simulated undetected error probability to the bounds produced by the exclusion and construction methods. We consider the CRC code $p(x) = x^3 + x + 1$ concatenated with the memory size $\nu = 4$ convolutional code with generator polynomial $(23, 35)_8$ in octal and $d_{\text{free}} = 7$. The information length is $k = 1021$ bits and thus the CRC codeword length is $n = 1024$ bits. The FER of this original convolutional code is plotted as a reference. The equivalent catastrophic convolutional code is $(255, 317)_8$ and its FER is also simulated.

We can see from Fig. 4.5 that the undetected error probability is upper bounded by the FER of the equivalent code, which equals P_{UD}, the probability of having an undetectable error in the CRC codeword, and this bound gets tighter as SNR increases. The equivalent code FER is above the probability of undetected error because it is possible that a frame has *both* an undetectable error event and a detectable error event, which causes a frame error in the equivalent code but does not cause an undetected error in the concatenated CRC and convolutional codes.

The upper bounds of P_{UD} are computed using the exclusion method (4.7) and construction method (4.15). In our calculation, the search depth limit of the exclusion method is $\tilde{d} = 14$, and $\hat{d} = 20$ is the depth limit for the construction method. Since $d_{\text{free}} = 7$, only undetectable single and double errors are considered. The probability sum of all large-distance terms given by (4.10c) are plotted to verify that they are negligible, except for the exclusion method at SNR below 0.75 dB. Although the construction method used a deeper search, it is still quite close to the exclusion method even in the low SNR region. It can be seen that these analysis methods deliver accurate bounds at high SNR for both the undetected error probability and the FER of the catastrophic code.

4.3 Search Procedure for Optimal CRC Codes

In this section, we will present an efficient way to find the optimal degree-m CRC code for a targeted convolutional code and information length k. Note that the performance of a CRC code depends on the information length [KC04]. A CRC code may be powerful for short sequences but have numerous undetectable long errors that are produced by a specific convolutional code.

Since x^m and x^0 terms are both one, there are 2^{m-1} candidates of degree-m CRC generator polynomials $p(x)$. In principle, either the exclusion method or the construction method can produce the undetected error probability for each candi-

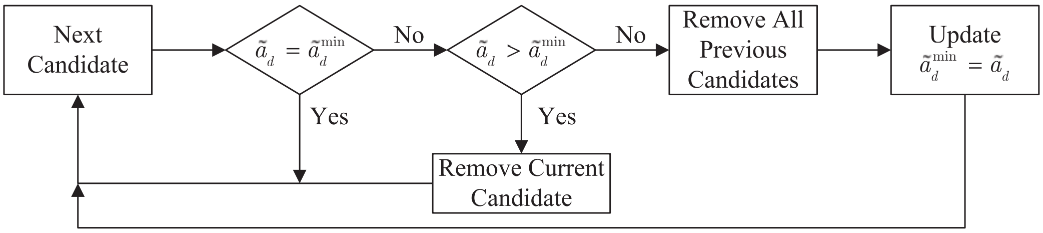

Next Candidate
$\tilde{a}_d = \tilde{a}_d^{\min}$
No
Yes
$\tilde{a}_d > \tilde{a}_d^{\min}$
No
Yes
Remove All Previous Candidates
Update $\tilde{a}_d^{\min} = \tilde{a}_d$
Remove Current Candidate

Figure 4.6: The work flow of keeping the CRC polynomials with the fewest undetectable errors with distance d.

date allowing selection of the best $p(x)$. However, this process is time-consuming if m is large. Both exclusion and construction methods need to compute the distance spectrum of undetectable errors up to some distance d. We can reduce the computation time by skipping the distance spectrum searches of suboptimal CRC codes.

When the FER is low, the undetectable error rate of a CRC code is dominated by the undetectable errors with the smallest distance. Let the smallest distance of the undetectable errors be d_{min}. We can evaluate a CRC-convolutional concatenated code by its distance spectrum at around d_{min}. To be more precise, a polynomial should be removed from the candidate list if it has a smaller d_{min} than the others or if it has more undetectable errors associated to the same d_{min}. In a convolutional code, since the number of error events a_d grows exponentially as the associated distance d increases, the cost to find all undetectable errors grows exponentially as well. Hence, the CRC code search starts with $d = d_{\text{free}}$ and updates the candidate list by keeping only the CRC generator polynomials with the fewest undetectable errors. Next, repeat the procedure with the next higher d until only one polynomial remains in the list. The routine of filtering the CRC polynomials at distance d can be performed by following the work flow in Fig. 4.6, where $\tilde{a}_d$ is the number of distance-d undetectable errors of the current CRC-convolutional concatenated code with certain information length, and $\tilde{a}_d^{\text{min}}$ is the smallest number of undetectable errors that has been found and have distance d. The initial value of $\tilde{a}_d^{\min}$ should be set to infinity for all d.

When $d < 2d_{\text{free}}$, only single errors are possible. The exclusion method can count the number of undetectable single errors of each candidate when $d \leq \tilde{d}$. We can perform a computer search for error events of the original code and check the divisibility of each of them. Note that if an error event is found to be undetectable, all of its possible offsets in the codeword should be counted. Since all candidates check the same set of error events, only one computer search with multiple divisibility checks (one for each CRC code) is sufficient. In contrast, using the construction method requires construction of equivalent encoders for each candidate separately. Hence, for the initial values of d near d_{free}, checking the divisibility via the exclusion method is preferred. Of course, once $d > \tilde{d}$, searching for undetectable single errors of the equivalent codes as in the construction method is the only approach.

When $d \geq 2d_{\text{free}}$, undetectable double errors need to be considered in addition to single errors. The divisibility test should be applied to all combinations of error event patterns $e_1(x)$, $e_2(x)$, and their gaps g_1. Even if the concept of cyclotomic cosets discussed in Appendix 4.A is utilized, we still need to construct all cyclotomic cosets through about 2^m divisions and also check if each of the remainder of $e_2(x)$ divided by $p(x)$ is in the same cyclotomic coset as the remainder of $e_1(x)$ divided by $p(x)$.

Alternatively, undetectable double errors can be directly created using the construction method. In the construction method, the error events connecting $\mathbf{S}^{\text{Z}}$ and states in $\mathbf{S}^{\text{D}}$ with distances between d_{free} and $d - d_{\text{free}}$ are found through computer searches. In fact, these events can be generated using the detectable error events of the original code previously found by exclusion if $d - d_{\text{free}} \leq \tilde{d}$. For example, since the detectable error pattern $e_1(x)$ is known, the corresponding error event in the equivalent encoder trellis starts from $\mathbf{S}^{\text{Z}}$ and traverses the trellis with the input sequence given by the quotient of $x^m e_1(x)$ divided by $p(x)$. The state $\mathrm{S}^{\mathrm{D}}_{\phi}$, where it ends, is thus determined by the last $m+\nu$ input bits, and its length $l^{\mathrm{ZD},\phi}_{d_1,i_1}$ has already been provided by the degree of $e_1(x)$.

For error events starting from states in $\mathbf{S}^{\mathrm{D}}$ and ending in $\mathbf{S}^{\mathrm{Z}}$ with pattern $e_2(x)$ previously obtained by exclusion, traverse the trellis in reverse from $\mathbf{S}^{\mathrm{Z}}$ with the input sequence given by the quotient of $x^{m+\nu+l'}e_2(x^{-1})$ divided by $x^m p(x^{-1})$, where $l' = l^{\mathrm{DZ},\psi}_{d_2,i_2} - 1$ is the degree of $e_2(x)$. Note that $x^{l'}e_2(x^{-1})$ and $x^m p(x^{-1})$ are the reverse bit-order polynomial representations of $e_2(x)$ and $p(x)$, respectively. The state $\mathrm{S}^{\mathrm{D}}_{\psi}$ where the error event begins is determined by the last $m+\nu$ input bits in reverse order.

An easier way to determine $\mathrm{S}^{\mathrm{D}}_{\psi}$, requiring no additional polynomial divisions, is presented in Appendix 4.C. Thus, to count the number of undetectable double errors, creating them directly as in the construction method is preferred.

According to the discussion at the end of Appendix 4.A, $d_{\min}$ is not likely to be much greater than $2d_{\mathrm{free}}$ when information length k is long enough. In other words, the CRC code search algorithm is usually finished before reaching $3d_{\mathrm{free}}$ and does not need to count the number of undetectable triple errors.

4.4 CRC Design Example for $\nu = 6$, $k = 1024$

As an example we present the best CRC codes of degree $m \leq 16$ specifically for the popular memory size $\nu = 6$ convolutional code with generator polynomial $(133, 171)_8$ with information length $k = 1024$ bits. Note that the proposed design method is applicable to all convolutional codes and information lengths, and not limited to the choices used for this example. The corresponding undetected error probability is also calculated and compared with existing CRC codes.

Table 4.2 and Table 4.3 show the standard CRC codes listed in [KC04] and the best CRC codes found by the search procedure in Section 4.3. For degrees with no standard codes, those recommended by Koopman and Chakravarty in

Table 4.2: Degree-3 to degree-9 standard CRC codes, CRC codes recommended by Koopman and Chakravarty [KC04], and best CRC codes for convolutional code $(133, 171)_8$ with information length $k = 1024$ bits.

	Gen.	Undetectable Single Distance Spectrum a_d^{ZZ}							
Name	Poly.	d	10	12	14	16	18	20	22
K&C-3	0x5		1	5	19	170	941	5050	29290
Best-3	0x7		0	7	24	169	879	5111	29363
CRC-4	0xF		1	2	11	79	464	2504	14719
Best-4	0xD		0	1	17	91	462	2537	14674
CRC-5	0x15		1	2	9	52	267	1378	8005
Best-5	0x11		0	0	4	52	230	1257	7275
CRC-6	0x21		0	1	4	21	124	572	3659
Best-6	0x29		0	0	1	22	124	641	3650
CRC-7	0x48		0	0	1	14	55	298	1877
Best-7	0x47		0	0	0	7	70	322	1867
CRC-8	0xEA		0	0	0	4	36	174	871
Best-8	0x89		0	0	0	1	29	177	938
K&C-9	0x167		0	0	0	4	13	73	477
Best-9	0x177		0	0	0	0	14	104	437
Original Distance Spectrum a_d			11	38	193	1331	7275	40406	234969

[KC04] are listed and called K&C. The notation of generator polynomials is in hexadecimal as used in [KC04]. For example, CRC-8 has generator polynomial $x^8+x^7+x^6+x^4+x^2+1$ expressed as 0xEA, where the most and least significant bits represent the coefficients of x^8 and x^1 terms, respectively. The coefficient of x^0 term is always one and thus omitted.

Table 4.2 and Table 4.3 also give the distance spectrum of undetectable single errors a_d^{ZZ} of each CRC code up to $d = 22$. The distance spectrum of the original

Table 4.3: Degree-10 to degree-16 standard CRC codes, CRC codes recommended by Koopman and Chakravarty [KC04], and best CRC codes for convolutional code $(133, 171)_8$ with information length $k = 1024$ bits.

	Gen.	Undetectable Single Distance Spectrum a_d^{ZZ}							
Name	Poly.	d	10	12	14	16	18	20	22
CRC-10	0x319		0	0	0	1	8	41	239
Best-10	0x314		0	0	0	0	3	49	223
CRC-11	0x5C2		0	0	0	0	7	17	107
Best-11	0x507		0	0	0	0	0	24	113
CRC-12	0xC07		0	0	0	0	3	12	48
Best-12	0xA10		0	0	0	0	0	4	66
K&C-13	0x102A		0	0	0	0	1	7	36
Best-13	0x1E0F		0	0	0	0	0	1	29
K&C-14	0x21E8		0	0	0	0	1	2	15
Best-14	0x314E		0	0	0	0	0	0	11
K&C-15	0x4976		0	0	0	0	1	1	6
Best-15	0x604C		0	0	0	0	0	0	3
CRC-16	0xA001		0	0	0	0	0	1	3
Best-16	0x8E61		0	0	0	0	0	0	1
Original Distance Spectrum a_d			11	38	193	1331	7275	40406	234969

convolutional code a_d is given as a reference. Note that, since this convolutional code has $d_{\mathrm{free}} = 10$, a smaller a_{20}^{ZZ} or a_{22}^{ZZ} does not mean fewer undetectable errors at distance $d = 20$ or 22. Undetectable double errors should also be counted for $d \geq 20$ to judge a candidate.

During the search for the best CRC codes with degrees $m \leq 11$, only single errors need to be considered because one candidate will outperform all the others before looking at $d = 20$. Although the best degree-11 CRC code has $d_{\mathrm{min}} = 20$, all the other candidates have $d_{\min} < 20$ and are eliminated before the end of the $d = 18$ round. Since the lengths of single errors $l_{d,i}$ for $d < 20$, ranging from 7 to 43, are much shorter than $n + \nu$, a candidate that has fewer types of dominant undetectable error events will have fewer dominant undetectable errors in total. In other words, when undetectable single errors dominate and information length k is long enough, the best CRC code should possess the smallest a_d^{ZZ}.

When $d_{\min} \geq 2d_{\mathrm{free}}$, the dominant undetectable errors include double errors. In this case, a smaller a_d^{ZZ} does not mean a better code because it only considers single errors. For example, the degree-16 polynomials 0xF8F1 and 0x8E61 both have $d_{\min} = 22$. The former has $a_{22}^{\mathrm{ZZ}} = 0$ while the latter has $a_{22}^{\mathrm{ZZ}} = 1$. However, at $d = 22$, the former has so many (2860) undetectable double errors that the number is greater than the total count of undetectable single and double errors $(1011 + 1424)$ of the latter, when the information length is $k = 1024$ bits.

Furthermore, if two candidates have the same $a_{d_{\min}}^{\mathrm{ZZ}}$, the information length k can impact the choice of the best CRC code. For example, the degree-16 polynomials 0x90DB and 0x8E61 both have $d_{\min} = 22$ and $a_{22}^{\mathrm{ZZ}} = 1$. 0x90DB has one length-44 dominant undetectable single error, but the longest error of 0x8E61 is length 36. Thus 0x90DB has eight less undetectable single errors at $d = 22$ than the polynomial 0x8E61, 0x90DB is not better at this specific information length ($k = 1024$) because it has more dominant undetectable double errors: at $d = 22$, 0x90DB has 1505 undetectable double errors containing eight combinations of $e_1(x)$ and $e_2(x)$ with combined lengths ranging from 694 to 1023 while 0x8E61 has 1424 undetectable double errors comprising only three combinations with relatively shorter lengths ranging from 405 to 754. Of course, other longer undetectable double errors, whose lengths are greater than $n+\nu$, are not counted. However, when the information length is shorter than 1010 bits, the polynomial 0x90DB has smaller total count of undetectable errors at $d = 22$ and is preferred.

In Fig. 4.7, the bounds of undetected error probability of the existing and

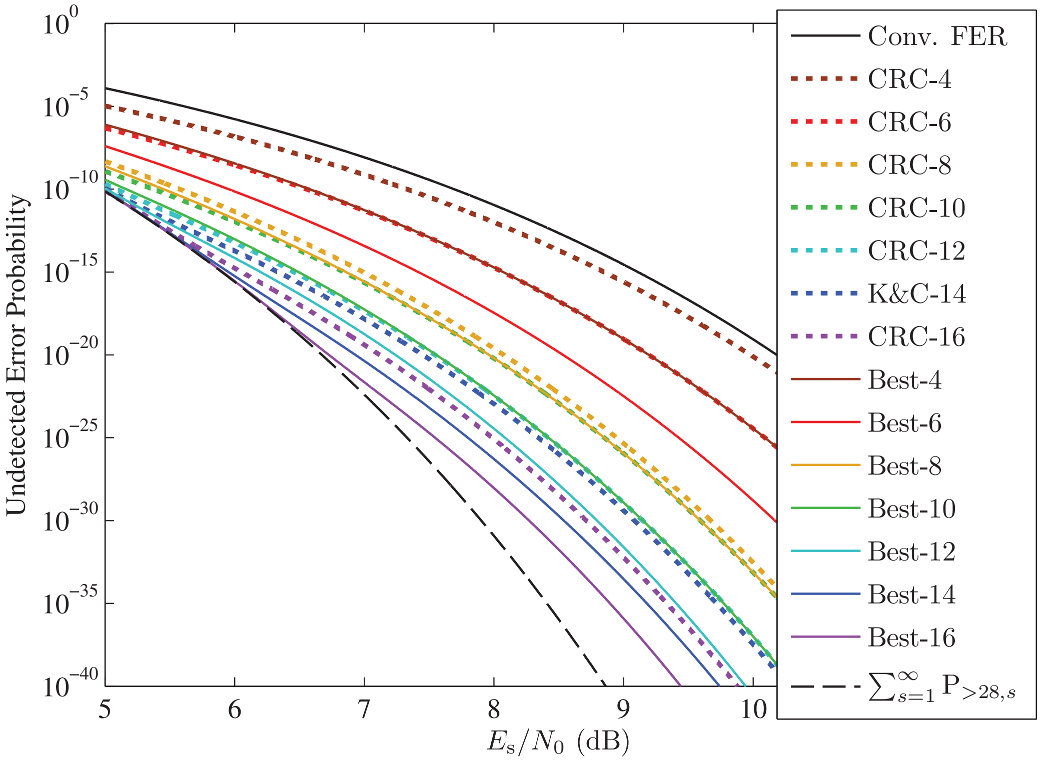


Figure 4.7: The undetected error probability of the existing and best CRC codes for convolutional code $(133, 171)_8$ with information length $k = 1024$ bits computed using the construction method.

best CRC codes are shown. For clarity, only even degrees of the CRC codes are displayed. The upper bound of the original convolutional code FER without any CRC code, calculated using transfer function techniques [LD12], is plotted as a reference. In our calculation, the search depth limit of the exclusion method is $\tilde{d} = 22$ and not enough for high degree CRC codes. Therefore, the construction method (4.15) was used with $\hat{d} = 28$.

The probability sum of all large-distance terms calculated using (4.10c) is also plotted to illustrate that the large-distance terms really are negligible even assuming they are all undetectable, except for the best degree-16 CRC code at SNR below 7 dB and some other codes at SNR below 6 dB. Note that since the operation $e^{\bar{\mathrm{P}}} - 1$ in (4.10c) causes non-negligible rounding errors in the high

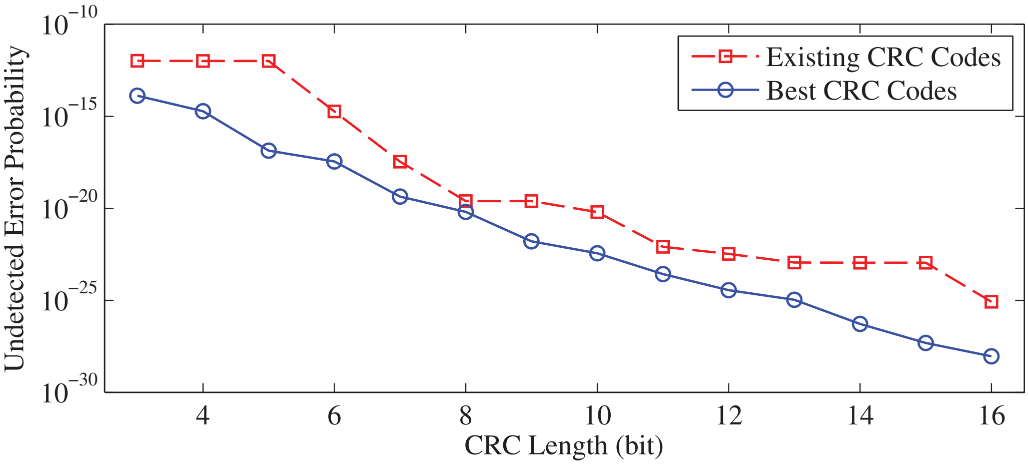


Figure 4.8: The undetected error probability of the existing and best CRC codes for convolutional code $(133, 171)_8$ with information length $k = 1024$ bits at SNR $= 8$ dB.

SNR region, it was approximated by the first ten terms, which results a similar expression as (4.10b) but only carried out for $1 \leq s \leq 10$.

Since $\hat{d} < 3d_{\text{free}}$, it is not necessary to evaluate triple or higher order errors under this truncation. It is clearly seen from the figure that the best degree-m CRC codes found by our procedure outperform the existing degree-m codes for all m. Furthermore, their performance is either better than or similar to the existing degree-$(m + 2)$ code except for $m = 6$. In other words, the proposed design can typically save 2 check bits while keeping the same error detection capability.

We can compare the error detection capability of all codes at a fixed SNR. If we draw a vertical line at SNR $= 8$ dB on Fig. 4.7, the intersections can be plotted along with the associated CRC lengths m in Fig. 4.8. The largest reduction of undetected error probability is about five orders of magnitude at $m = 5$ where the existing CRC code has an undetected error probability of 1.008×10^{-12} and the newly designed CRC code has an undetected error probability of 1.360×10^{-17}. Note that these numbers are calculated at 8 dB SNR, and the reduction gets more significant as SNR increases.

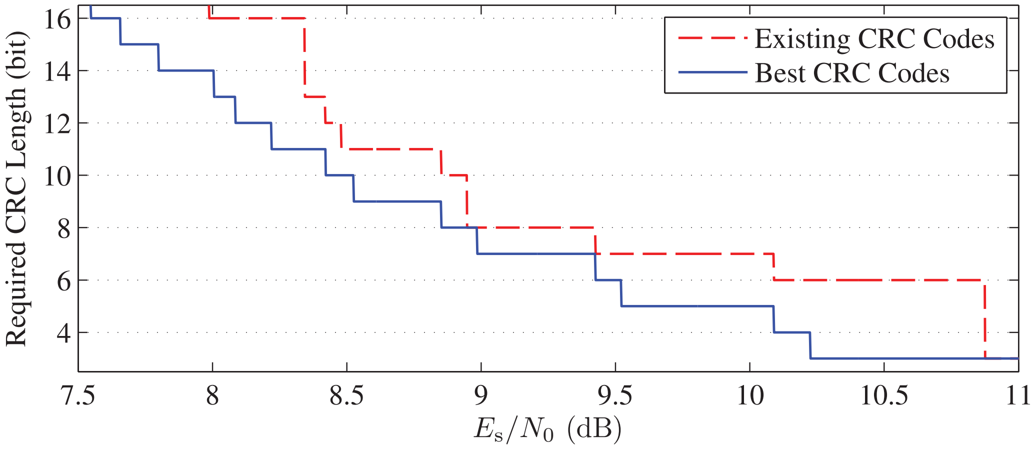


Figure 4.9: The required CRC lengths for the existing and best CRC codes for convolutional code $(133, 171)_8$ with information length $k = 1024$ bits to achieve undetected error probability below 10^{-25}.

Since the existing CRC codes are not tailored to the convolutional code, a higher degree code does not necessarily have a better performance. We observe that there are three almost horizontal region for the existing CRCs with degrees m ranging from 3 to 5, from 8 to 9, and from 13 to 15. The reason is that the codes in each region have similar number of dominant undetectable errors as can be seen in Table 4.2 and Table 4.3. In contrast, the CRC codes designed using our procedure show steady improvement as the degree increases.

The existing and best CRC codes can also be compared in terms of the required CRC length to achieve certain undetected error probability. Assume our target is to reach undetected error probability below 10^{-25}. This can be shown by drawing a horizontal line at the target probability level on Fig. 4.7 and plotting the CRC lengths associated to the crossed points as a function of SNR as in Fig. 4.9. For most of the SNR levels, the best CRC codes requires two fewer check bits than the existing CRC codes to achieve the same error detection capability. At SNR around 10.5 dB, the best CRC code requires three fewer check bits than the existing CRC code. Since the existing code required six check bits, this is a 50% reduction.

4.5 Conclusion

A good CRC code in a convolutionally coded system should minimize the undetected error probability. To calculate this probability, two methods based on distance spectrum are proposed. The exclusion method starts with all possible single and multiple error patterns of the convolutional code and excludes them one by one by testing if they are detectable. In the construction method, undetectable errors are mapped to error events of an equivalent convolutional code, which is the combination of the CRC code and the original convolutional code. The computer search for error events in the construction method does not need to record the error patterns and thus can go deeper than the search in the exclusion method. However, the construction method could encounter difficulties while dealing with high-degree CRC codes. Moreover, the construction method is generally applicable to the performance analysis of catastrophic convolutional codes.

We also propose a search procedure to identify the best CRC codes for a specified convolutional encoder and information length. A candidate CRC code is excluded if it has more low-distance undetectable errors. Therefore, the best CRC polynomial is guaranteed to have the fewest dominant undetectable errors and minimizes the probability of undetected error when SNR is high enough. When undetectable double errors dominate, the choice of the best CRC polynomial is more dependent on the information length. In an example application of the design procedure for the popular 64-state convolutional code with information length $k = 1024$, new CRC codes provided significant reduction in undetected error probability compared to the existing CRC codes with the same degrees. With the proposed design, we are able to save two check bits in most cases while having the same error detection capability.

4.A Efficient Search for Undetectable Double Errors

In (4.4) and (4.5), the indicator function is evaluated for every g_1 given $e_1(x)$ and $e_2(x)$. However, there is a more efficient way to check the divisibility and can save a lot of computation time when $n + \nu$ is large.

The remainder of any polynomial in $\mathbf{GF}(2)[x]$ divided by $p(x)$ forms a quotient ring $\mathbf{GF}(2)[x]/p(x)$. Two polynomials mapped to the same element of the quotient ring are called *congruent*. If $x^{g_1+l_2}e_1(x)$ is congruent to $e_2(x)$ modulo $p(x)$, then their combination forms an undetectable double error. To characterize the remainder of $x^{g_1+l_2}e_1(x)$ modulo $p(x)$, we apply the concept of cyclotomic coset[Lin04], which is originally defined for integers, to polynomials. For polynomials in $\mathbf{GF}(2)[x]$, define x-cyclotomic coset modulo $p(x)$ containing $e(x)$ as

$$\mathbf{C}_{e(x)} = \left\{ x^h e(x) \,(\mathrm{mod}\ p(x)) \middle| \, h = 0, 1, \cdots \right\}, \tag{4.16}$$

which includes the remainder of all possible offsets of $e(x)$ divided by $p(x)$. One can verify that two cyclotomic cosets are either identical or disjoint, so all of the distinct x-cyclotomic cosets modulo $p(x)$ form a partition of quotient ring $\mathbf{GF}(2)[x]/p(x)$. For example, consider a degree-2 primitive polynomial $p(x) = x^2 + x + 1$, and we have $\mathbf{C}_0 = \{0\}$ and $\mathbf{C}_1 = \{1, x, x + 1\}$ forming a partition of $\mathbf{GF}(2)[x]/p(x)$; consider a degree-2 non-primitive polynomial $p(x) = x^2 + 1$, and we have $\mathbf{C}_0 = \{0\}$, $\mathbf{C}_1 = \{1, x\}$, and $\mathbf{C}_{x+1} = \{x + 1\}$ forming a partition of $\mathbf{GF}(2)[x]/p(x)$. In both cases, $\mathbf{C}_0$ is trivial since it only contains the "zero" element. In the primitive case, $\mathbf{C}_1$ is the only non-trivial cyclotomic coset and its cardinality is $|\mathbf{C}_1| = |\mathbf{GF}(2)[x]/p(x)| - 1 = 2^m - 1$. In the non-primitive case, there are multiple non-trivial cyclotomic cosets and their sizes are smaller than $2^m - 1$. In fact, there is only one unique non-trivial cyclotomic coset if $p(x)$ is a primitive polynomial.

It is obvious that if $e_1(x)$ and $e_2(x)$ belong to different cyclotomic cosets, there is no way to have a g_1 that makes $x^{g_1+l_2}e_1(x)$ congruent to $e_2(x)$ modulo $p(x)$. In other words, it is unnecessary to check whether any g_1 creates an undetectable double error with the specific $e_1(x)$ and $e_2(x)$. If $e_1(x)$ and $e_2(x)$ belong to the same cyclotomic coset $\mathbf{C}_{e_1(x)}$, only one proper $g_1 \in \left[0, \left|\mathbf{C}_{e_1(x)}\right| - 1\right]$ can make $x^{g_1+l_2}e_1(x) + e_2(x)$ divisible by $p(x)$. Denote this particular g_1 as g_1'. Once we find g_1', all possible g_1 that create undetectable double errors are just $g_1 = g_1' + u\left|\mathbf{C}_{e_1(x)}\right|$ for non-negative integer u satisfying $g_1 + l_1 + l_2 \leq n + \nu$.

Note that when $e_1(x) = e_2(x)$, they will belong to the same cyclotomic coset no matter what CRC generator polynomial $p(x)$ is. That is to say, no CRC code is able to detect such double error if these two error events have a proper gap g_1. Fortunately, the smallest proper gap is $g_1' = -l_2 \left(\bmod \; \left|\mathbf{C}_{e_1(x)}\right|\right)$ so the total length of the undetectable double error is $\left|\mathbf{C}_{e_1(x)}\right| + l_1$. Hence, when $n+\nu$ is small enough, such double error will never occur. On the other hand, when $n + \nu$ is large, d_{min}, which is the shortest distance of the undetectable errors, will be upper bounded by $2d_{\text{free}}$.

4.B The Relationship between $\mathbf{\Delta}_i$ and $\mathbf{C}_{e(x)}$

Define the state of the equivalent code at time $n-g$ as $q_g'(x)$, which is a polynomial with maximum degree $m + \nu - 1$ representing consecutive $m + \nu$ bits from the $x^{g+m+\nu-1}$ term to the x^g term in $x^m q'(x)$ for $g \in [-\nu, n]$. The corresponding state of the original code is given by the coefficients of the terms from $x^{m+\nu-1}$ to x^m in polynomial $q_g'(x)p(x)$. Let $q_{g,u}'$ and p_u be the coefficients of x^u in $q_g'(x)$ and $p(x)$, respectively. Then the coefficient of x^u for $u \in [m, m + \nu - 1]$ in $q_g'(x)p(x)$ is given by

$$\sum_{v=0}^{m} q_{g,u-v}' \, p_v. \tag{4.17}$$

If the state of the original code is known and $q_{g,v}'$ is known for $v \in [\nu, m + \nu - 1]$, $q_{g,\nu-1}'$ can be solved through (4.17) when $u = m+\nu-1$ because $p_m = 1$. Moreover, the rest of $q_{g,v}'$ for $v = \nu - 2, \cdots, 1, 0$ can be solved one by one in the same way.

Assume that the all-zero codeword is sent, i.e. $q(x) = 0$, and the trellis path enters a detectable-zero state $\mathrm{S}_i^{\mathrm{D}}$ at time $n - g$. Also, let $e(x)$ be a polynomial of degree smaller than or equal to $n - g - 1$ representing the length-$(n - g)$ input sequence of the original convolutional encoder from the beginning to time $n - g$, and it is given by the coefficients from the x^{n-1} term to the x^g term in $q'(x)p(x)$. Since the state at time $n - g$ is $\mathrm{S}_i^{\mathrm{D}}$, $e(x)$ must be non-divisible by $p(x)$.

The remainder of $e(x)$ divided by $p(x)$ is given by

$$e(x) \pmod{p(x)} = \sum_{u=0}^{m-1} x^u \sum_{v=u+1}^{m} q'_{g,m+u-v} p_v, \tag{4.18}$$

which is totally governed by $q'_g(x)$, or $\mathrm{S}_i^{\mathrm{D}}$. If the remainder is known, the bits $q'_{g,v}$ for $v \in [0, m-1]$ can be solved uniquely through back substitution for $u = m-1, m-2, \cdots, 0$ because $p_m = 1$. Furthermore, the whole polynomial $q'_g(x)$, or $\mathrm{S}_i^{\mathrm{D}}$, can be solved by letting (4.17) equal to zero for $u = m + \nu - 1, m + \nu - 2, \cdots, m$ because the state of the original code is just ν zeros. Hence, the remainder of $e(x)$ divided by $p(x)$ determines a detectable-zero state $\mathrm{S}_i^{\mathrm{D}}$, and vice versa. Furthermore, each of the $2^m - 1$ non-zero elements in $\mathbf{GF}(2)[x]/p(x)$ corresponds to a unique state in $\mathbf{S}^{\mathrm{D}}$.

To find all states in $\mathbf{\Delta}_i$, we can specify $q'_{g-h,0}$, the input bit to the equivalent encoder at time $n - g + h$, for $h = 1, 2, \cdots$ such that the input bits to the original convolutional encoder after time $n - g$ are all zeros and thus the following states are in $\mathbf{S}^{\mathrm{D}}$. By doing so, we know that the polynomials $q'_{g-h}(x)$ will represent the states in $\mathbf{\Delta}_i$. This procedure is finished when certain $q'_{g-h}(x)$ represents $\mathrm{S}_i^{\mathrm{D}}$ again. During this procedure, the corresponding input sequence to the original encoder from the beginning to time $n - g + h$ is simply $x^h e(x)$ because the input bits after time $n - g$ are all zeros. By the definition given in (4.16), the remainder of $x^h e(x)$ divided by $p(x)$ is an element of $\mathbf{C}_{e(x)}$. In addition, we know that this remainder corresponds to a state in $\mathbf{\Delta}_i$. Therefore, $\mathbf{\Delta}_i$ and $\mathbf{C}_{e(x)}$ contain the same elements but just represented in different forms.

4.C An Easier Search for Error Events Not Starting from $\mathbf{S}^{\mathrm{Z}}$

In the construction method, we need to search for single error events starting from a state in $\mathbf{S}^{\mathrm{D}}$ and ending at $\mathbf{S}^{\mathrm{Z}}$ or a state in $\mathbf{S}^{\mathrm{D}}$ in order to construct undetectable multiple errors. Performing computer searches for all of them is definitely an approach, but we could possibly save a huge amount of computation by utilizing the search result of single error events starting from $\mathbf{S}^{\mathrm{Z}}$. Since these searches all target on error events of the original code but just have different initial states, one search result has already provided enough information to characterize these error events. Therefore, with the knowledge of the relationship between these initial states, one search result is applicable to the others.

Let $q''_j(x)$ be a polynomial of degree smaller than or equal to $m + \nu - 1$ representing the state $\mathrm{S}^{\mathrm{D}}_j$, and $q''_0(x) = 0$ represent $\mathbf{S}^{\mathrm{Z}}$. Assume that a known single error event with length l starts from $\mathbf{S}^{\mathrm{Z}}$ and ends at the state represented by $q''_\phi(x)$. Note that this error event could be either detectable or undetectable. If this error event starts from a detectable-zero state $\mathrm{S}^{\mathrm{D}}_\psi$ instead of $\mathbf{S}^{\mathrm{Z}}$, its length is unchanged and we just need to know where it ends.

We can treat the error pattern at the convolutional encoder as an input to a discrete time system and $q'(x)$ as its output. Since this system is linear, the end state, given by a subsequence of $q'(x)$, is determined by the superposition of the zero state response and the zero input response. Apparently, $q''_\phi(x)$ is the zero state response because it is the result when the error event starts at $\mathbf{S}^{\mathrm{Z}}$. The zero input response is governed by its initial state $\mathrm{S}^{\mathrm{D}}_\psi$. Let $\mathrm{S}^{\mathrm{D}}_\theta$ be the end state of a length-l trellis path starting from $\mathrm{S}^{\mathrm{D}}_\psi$ while keeping the error pattern all zeros. We can find θ because it satisfies $\mathrm{S}^{\mathrm{D}}_\theta \in \mathbf{\Delta}_\psi$ and $\delta_{\psi,\theta} = l \pmod{|\mathbf{\Delta}_\psi|}$. Hence, the end state of the error event starting from $\mathrm{S}^{\mathrm{D}}_\psi$ is represented by $q''_\phi(x) + q''_\theta(x)$.

If we want to find the initial state $\mathrm{S}^{\mathrm{D}}_\psi$ such that the detectable single error event ends at $\mathbf{S}^{\mathrm{Z}}$, the same idea can still be applied. Since the end state is $\mathbf{S}^{\mathrm{Z}}$, the zero state and zero input responses should cancel each other, i.e. $\theta = \phi$. Therefore, we can find ψ because it satisfies $\mathrm{S}_{\psi}^{\mathrm{D}} \in \mathbf{\Delta}_{\phi}$ and $\delta_{\phi,\psi} = -l \pmod{|\mathbf{\Delta}_{\phi}|}$.

CHAPTER 5

Conclusion

In the first part of the dissertation, we describe two analytical PER prediction methods for punctured convolutional codes with a uniform random interleaver. The analytical methods are based on the codes' transfer functions and the modeling of the distribution of half of log-error-likelihood ratio (HLELR), or essentially LLR. The Q-function approximation method approximates the HLELR using a mixture of normal distributions and relies on the Gaussian Q-function approximation. The saddlepoint approximation method models the HLELR as a mixture of folded normal distributions and applies the saddlepoint approximation. Unlike the heuristic methods, the proposed methods require absolutely no curve-fitting or parameter calibrations. Furthermore, compared to the existing heuristic methods, they deliver more accurate predictions for SISO systems in AWGN and Rayleigh fading channels, as well as MIMO OFDM systems in frequency selective fading channels.

We then generalize the result of the analytical PER prediction for systems with a uniform random interleaver to systems with a repeated-pattern interleaver. In a punctured convolutionally coded MIMO OFDM system employing a repeated-pattern interleaver, the LLRs are no longer i.i.d. random variables. The distribution of a bit's LLR depends on its location in the packet. Furthermore, two LLRs can be highly correlated if they belong to the same subcarrier, whether they come from the same spatial stream or not. With the consideration of LLRs' correlation, the proposed Q-function approximation method significantly outperforms all the other methods.

In the second part of the dissertation starting in Chapter 4, we consider a system employing a CRC code concatenated with a convolutional code. To calculate the undetected error probability of this system, two methods based on distance spectrum are proposed. The exclusion method starts with all possible single and multiple error patterns of the convolutional code and excludes them one by one by testing if they are detectable. In the construction method, undetectable errors are mapped to error events of an equivalent convolutional code, which is the combination of the CRC code and the original convolutional code. The computer search for error events in the construction method does not need to record the error patterns and thus can go deeper than the search in the exclusion method. However, the construction method could encounter difficulties while dealing with high-degree CRC codes. Moreover, the construction method is generally applicable to the performance analysis of catastrophic convolutional codes.

We also propose a search procedure to identify the best CRC codes for a specified convolutional encoder and information length. A candidate CRC code is excluded if it has more low-distance undetectable errors. Therefore, the best CRC polynomial is guaranteed to have the fewest dominant undetectable errors and minimizes the probability of undetected error when SNR is high enough. When undetectable double errors dominate, the choice of the best CRC polynomial is more dependent on the information length. In an example application of the design procedure for the popular 64-state convolutional code with information length $k = 1024$, new CRC codes provided significant reduction in undetected error probability compared to the existing CRC codes with the same degrees. With the proposed design, we are able to save two check bits in most cases while having the same error detection capability.

References

[AAS10] A. Alvarado, E. Agrell, L. Szczecinski, and A. Svensson. "Exploiting UEP in QAM-based BICM: interleaver and code design." **58**(2):500–510, Feb. 2010.

[ASA11] A. Alvarado, L. Szczecinski, and E. Agrell. "On the Performance of BICM with Trivial Interleavers in Nonfading Channels." In *Proc. 2011 IEEE Int. Conf. Commun. (ICC)*, pp. 1–6, Jun. 2011.

[ASF09] A Alvarado, L. Szczecinski, R. Feick, and L. Ahumada. "Distribution of L-values in Gray-mapped M^2-QAM: closed-form approximations and applications." *IEEE Trans. Commun.*, **57**(7):2071–2079, Jul. 2009.

[Azz85] Adelchi Azzalini. "A class of distributions which includes the normal ones." *Scandinavian J. Stat.*, pp. 171–178, 1985.

[BHP90] G. Begin, D. Haccoun, and C. Paquin. "Further results on high-rate punctured convolutional codes for Viterbi and sequential decoding." *IEEE Trans. Commun.*, **38**(11):1922–1928, Nov. 1990.

[Bil97] Jeff A Bilmes. "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models." Tr-97-021, Int. Comput. Sci. Inst. (ICSI), 1997.

[BM09] Nicola Baldo and Marco Miozzo. "Spectrum-aware Channel and PHY layer modeling for ns3." In *Proc. 4th Int. ICST Conf. Performance Evaluation Methodologies and Tools*, p. 2. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[CBH93] G. Castagnoli, S. Brauer, and M. Herrmann. "Optimization of cyclic redundancy-check codes with 24 and 32 parity bits." *IEEE Trans. Commun.*, **41**(6):883–892, Jun. 1993.

[CCG79] J. Cain, G. Clark, and J.M. Geist. "Punctured convolutional codes of rate $(n-1)/n$ and simplified maximum likelihood decoding." *IEEE Trans. Inf. Theory*, **25**(1):97–100, Jan. 1979.

[CDS03] M. Chiani, D. Dardari, and Marvin K. Simon. "New exponential bounds and approximations for the computation of error probability in fading channels." *IEEE Trans. Wireless Commun.*, **2**(4):840–845, Jul. 2003.

[CJ89] M.L. Cedervall and R. Johannesson. "A fast algorithm for computing distance spectrum of convolutional codes." *IEEE Trans. Inf. Theory*, **35**(6):1146–1159, Nov. 1989.

[CTB98] G. Caire, Giorgio Taricco, and Ezio Biglieri. "Bit-interleaved coded modulation." *IEEE Trans. Inf. Theory*, **44**(3):927–946, May 1998.

[DCH10] R.C. Daniels, C.M. Caramanis, and R.W. Heath. "Adaptation in Convolutionally Coded MIMO-OFDM Wireless Systems Through Supervised Learning and SNR Ordering." *IEEE Trans. Veh. Technol.*, **59**(1):114–126, Jan. 2010.

[Erc04] Vinko Erceg et al. "TGn channel models." 03/940r4, IEEE 802.11, May 2004.

[Eri03] Ericsson. "System-level evaluation of OFDM further considerations." TSG-RAN WG1 #35 R1-031303, 3GPP, Nov. 2003.

[EWD14] E. Eraslan, Chao-Yi Wang, and B. Daneshrad. "Practical Energy-Aware Link Adaptation for MIMO-OFDM Systems." *IEEE Trans. Wireless Commun.*, **13**(1):246–258, Jan. 2014.

[GB09] Alan Genz and Frank Bretz. *Computation of multivariate normal and t probabilities*, volume 45. Springer, 2009.

[GL13] M. Ghosh and F. LaSita. "Puncturing of CRC Codes for IEEE 802.11ah." In *Proc. 2013 IEEE 78th Veh. Technology Conf. (VTC Fall)*, pp. 1–5, Sep. 2013.

[HB89] D. Haccoun and G. Begin. "High-rate punctured convolutional codes for Viterbi and sequential decoding." *IEEE Trans. Commun.*, **37**(11):1113–1125, Nov. 1989.

[HSP01] R.W. Heath, S. Sandhu, and A Paulraj. "Antenna selection for spatial multiplexing systems with linear receivers." *IEEE Commun. Lett.*, **5**(4):142–144, Apr. 2001.

[JKW10] T.L. Jensen, S. Kant, J. Wehinger, and B.H. Fleury. "Fast Link Adaptation for MIMO OFDM." *IEEE Trans. Veh. Technol.*, **59**(8):3766–3778, Oct. 2010.

[Kaz01] P. Kazakov. "Fast calculation of the number of minimum-weight words of CRC codes." *IEEE Trans. Inf. Theory*, **47**(3):1190–1195, Mar. 2001.

[KC04] P. Koopman and T. Chakravarty. "Cyclic redundancy code (CRC) polynomial selection for embedded networks." In *Proc. 2004 IEEE Int. Conf. Dependable Syst. and Networks (DSN)*, pp. 145–154, Jun. 2004.

[KD10] Hun Seok Kim and B. Daneshrad. "Energy-Constrained Link Adaptation for MIMO OFDM Wireless Communication Systems." *IEEE Trans. Wireless Commun.*, **9**(9):2820–2832, Sep. 2010.

[KL92] S. Kallel and C. Leung. "Efficient ARQ schemes with multiple copy decoding." *IEEE Trans. Commun.*, **40**(3):642–650, Mar. 1992.

[KL10] A Kenarsari-Anhari and L. Lampe. "An analytical approach for performance evaluation of BICM transmission over Nakagami-m fading channels." *IEEE Trans. Commun.*, **58**(4):1090–1101, Apr. 2010.

[KM84] T. Klove and M. Miller. "The Detection of Errors After Error-Correction Decoding." *IEEE Trans. Commun.*, **32**(5):511–517, May 1984.

[Koo02] P. Koopman. "32-bit cyclic redundancy codes for Internet applications." In *Proc. 2002 IEEE Int. Conf. Dependable Syst. and Networks (DSN)*, pp. 459–468, 2002.

[KS05] R. Khalili and K. Salamatian. "A new analytic approach to evaluation of packet error rate in wireless networks." In *Proc. 3rd IEEE Annu. Conf. Commun. Networks and Services Research (CNSR)*, pp. 333–338, May 2005.

[LBF79] S. Leung-Yan-Cheong, Earl R. Barnes, and D. Friedman. "On some properties of the undetected error probability of linear codes." *IEEE Trans. Inf. Theory*, **25**(1):110–112, Jan. 1979.

[LC05] Ren-Der Lin and Wen-Shyen Chen. "Fast calculation algorithm of the undetected errors probability of CRC codes." In *Proc. 2005 IEEE 19th Int. Conf. Advanced Inform. Networking and Applicat. (AINA)*, volume 2, pp. 480–483, Mar. 2005.

[LD12] Chung-Yu Lou and B. Daneshrad. "PER prediction for convolutionally coded MIMO OFDM systems—An analytical approach." In *Proc. 2012 IEEE Military Commun. Conf. (MILCOM)*, pp. 1–6, Oct. 2012.

[Lin04] San Ling. *Coding theory: a first course.* Cambridge University Press, 2004.

[LK04] Jing Li and E. Kurtas. "Punctured convolutional codes revisited: the exact state diagram and its implications." In *Rec. 38th Asilomar Conf. Signals, Syst. and Comput. (ACSSC)*, volume 2, pp. 2015–2019 Vol.2, Nov. 2004.

[LNN61] F. C. Leone, L. S. Nelson, and R. B. Nottingham. "The Folded Normal Distribution." *Technometrics*, **3**(4):543–550, 1961.

[McE98] Robert J McEliece. "How to compute weight enumerators for convolutional codes." *Communications and Coding*, pp. 121–141, 1998.

[MF09] A. Martinez and A. Guillen i Fabregas. “Large-SNR error probability analysis of BICM with uniform interleaving in fading channels.” *IEEE Trans. Wireless Commun.*, **8**(1):38–44, Jan. 2009.

[MFC06] A Martinez, A Guillen i Fabregas, and G. Caire. “Error probability analysis of bit-interleaved coded modulation.” *IEEE Trans. Inf. Theory*, **52**(1):262–271, Jan. 2006.

[MHA14] M.T. Malik, M.J. Hossain, and M.-S. Alouini. “Analysis and Design of Generalized BICM-T System.” **62**(9):3223–3236, Sep. 2014.

[ML99] E. Malkamaki and H. Leib. “Evaluating the performance of convolutional codes over block fading channels.” *IEEE Trans. Inf. Theory*, **45**(5):1643–1646, Jul. 1999.

[NR98] S. Nanda and K.M. Rege. “Frame error rates for convolutional codes on fading channels and the concept of effective Eb/N0.” *IEEE Trans. Veh. Technol.*, **47**(4):1245–1250, Nov. 1998.

[PT87] M.B. Pursley and D.J. Taipale. “Error Probabilities for Spread-Spectrum Packet Radio with Convolutional Codes and Viterbi Decoding.” *IEEE Trans. Commun.*, **35**(1):1–12, Jan. 1987.

[PZR07] Fei Peng, Jinyun Zhang, and W.E. Ryan. “Adaptive Modulation and Coding for IEEE 802.11n.” In *Proc. 2007 IEEE Wireless Commun. Network Conf. (WCNC)*, pp. 656–661, Mar. 2007.

[SCH07] Shin-Lin Shieh, Po-Ning Chen, and Y. S Han. “Flip CRC Modification for Message Length Detection.” *IEEE Trans. Commun.*, **55**(9):1747–1756, Sep. 2007.

[SGL09] I Stupia, F. Giannetti, V. Lottici, and L. Vandendorpe. “A Novel Link Performance Prediction Method for Coded MIMO-OFDM Systems.” In *Proc. 2009 IEEE Wireless Commun. Network Conf. (WCNC)*, pp. 1–5, Apr. 2009.

[Sta09] “IEEE 802.11n-2009 part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification.”, Oct. 2009.

[SZS07] Krishna Sayana, Jeff Zhuang, and Ken Stewart. “Link performance abstraction based on mean mutual information per bit (MMIB) of the LLR channel.” C802.16m-07/097, IEEE 802.16 BWA WG, 2007.

[TB02] F. Tosato and P. Bisaglia. “Simplified soft-output demapper for binary interleaved COFDM with application to HIPERLAN/2.” In *Proc. 2002 IEEE Int. Conf. Commun. (ICC)*, volume 2, pp. 664–668 vol.2, 2002.

[TS03] S. Shawn Tsai and Anthony C. K. Soong. "Effective-SNR Mapping for Modeling Frame Error Rates in Multiple-state Channels." C30-20030429-010, 3GPP2, Apr. 2003.

[TWS08] Peng Tan, Yan Wu, and Sumei Sun. "Link adaptation based on adaptive modulation and coding for multiple-antenna OFDM system." *IEEE J. Sel. Areas Commun.*, **26**(8):1599–1606, Oct. 2008.

[Vit71] A.J. Viterbi. "Convolutional Codes and Their Performance in Communication Systems." *IEEE Trans. Commun. Technol.*, **19**(5):751–772, Oct. 1971.

[Vit98] A.J. Viterbi. "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes." *IEEE J. Sel. Areas Commun.*, **16**(2):260–264, Feb. 1998.

[VO09] Andrew J Viterbi and Jim K Omura. *Principles of digital communication and coding.* Courier Dover Publications, 2009.

[WTA06] Lei Wan, Shiauhe Tsai, and M. Almgren. "A fading-insensitive performance metric for a unified link quality model." In *Proc. 2006 IEEE Wireless Commun. Network Conf. (WCNC)*, volume 4, pp. 2110–2114, Apr. 2006.

[YZS06] Ping-Cheng Yeh, S.A Zummo, and W.E. Stark. "Error probability of bit-interleaved coded modulation in wireless environments." *IEEE Trans. Veh. Technol.*, **55**(2):722–728, Mar. 2006.