Channel Matched Iterative Decoding for Magnetic Recording Systems

A DISSERTATION

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

AbdelHakim Salem Alhussien

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Professor Jaekyun Moon, Advisor

April 2009

Acknowledgements

I will always be in debt to my PhD adviser, Professor Jaekyun Moon, who introduced me to the area of signal processing and coding for the magnetic recording channel. It is only through his immense support, deep intuition, and thorough insightful advise that this work is made possible. His mark in my academic, professional, and personal lives will never be forgotten.

I am also very grateful to Profs Tom Luo, and Nihar Jindal from the Electrical and Computer Engineering department and Prof Paul Garrett from the Mathematics department at the University of Minnesota for serving as committee members in my preliminary and final oral exams.

I acknowledge the enriching technical discussions I had with my friends and colleagues in the Communications and Data Storage (CDS) lab, including Jihoon Park, Farshid Rafiee, Jaewook Lee, Seongwook Jeong, and Daejung Yoon.

My experience at graduate school was made enjoyable through the good times I spent with my close friends at the university Minnesota, and through the encouragement of old friends who stood beside me over the years, for that, I will always be thankful.

My parents have sacrificed and given so much to bring me up and invest in my education, this thesis is a culmination of their love, support, and devotion through all my life, a debt I can never repay. I will always aspire to make them more proud. They are the beacon of my way. My sisters, Sumaya and Rowayda, have filled my childhood with unconditional love and overwhelming happiness that made me into what I am.

It is not spaces that connect the words of my thesis, rather, it is unlimited sacrifice and dedication of the one love of my life, my wife, Ruqayyah. I will ever cherish what she has endured to see this work accomplished. She has always believed in me and encouraged my every step. Living to her expectation is my true success and fulfillment. Ruqayyah, you are the meaning of my existence, my inspiration, and the driving force behind my success. This thesis is dedicated to you with all my eternal love.

To the love of my life...

the joy of my heart...

to my lady, Ruqayyah...

رُقَيَّة

Abstract

The perpendicular magnetic recording channel (PMRC) is corrupted by sever inter-symbol interference and data-dependent media noise, in addition to a variety of other bursty impairments. Thus far, the hard decodable symbol correcting Reed-Solomon (RS) code has been the industry standard for outer error control coding (ECC). This thesis proposes two novel ECC schemes in the migration toward next generation high density recording. The first scheme is a two-level concatenation of channel-matched turbo equalization (TE) and outer RS, replacing current inner parity correction codes. Conventional TE is matched to the channel via the incorporation of the error pattern correction code (EPCC), which works iteratively with the other constituent code in TE, whether block or convolutional, to suppress the occurrence of low-Euclidean-distance errors at the output of the channel detector. To understand this mechanism, and with no loss of generality, we derive the error Euclidean distance distribution of TE-EPCC for the Dicode channel, and show that EPCC substantially increases the interleaver gain exponent of low Euclidean weight errors. Furthermore, we derive an upper bound on the BER of TE-EPCC, and employ it to show that TE-EPCC delivers significant gains in the error floor and cliff regions compared to conventional precoded and unprecoded TE for a variety of channel conditions and code rates. The second proposed ECC system is a tensor product concatenation of EPCC and Q-ary LDPC (T-EPCC-QLDPC). This concatenation scheme enables the use of byte-long component EPCC without jeopardizing the overall code rate. Hence, the multiple error correction capability of EPCC is maintained at very low signal-to-noise ratios, while the component non-binary LDPC insures correct syndromes are available for the decoding of tensor symbols (EPCC code-blocks). We introduce a low complexity iterative soft decoder of T-EPCC-QLDPC, in which the component EPCC and QLDPC exchange multi-level log-likelihood ratios (mlLLR) that represent their beliefs on the reliability of error-syndromes. Moreover, we show that the two-level decoder provides a better performance-complexity tradeoff compared to single-level binary and Q-ary LDPC.

Contents

List of Tables

List of Figures

Acronyms

AFE	Analogue Front End
AWGN	Additive White Gaussian Noise
BCJR	Bahl Cocke Jelinek Raviv
BER	Bit Error Rate
BP	Belief Propagation
ECC	Error Control Code
EPCC	Error Pattern Correction Code
HDD	Hard Disc Drive
ISI	Inter-Symbol Interference
i.u.d.	Independent Uniformly Distributed
LDPC	Low Density Parity Check
LLR	Log Likelihood Ratio
mlLLR	Multilevel LLR
MAP	Maximum A posteriori Probability
ML	Maximum Likelihood
NRZ	Non-return to Zero
PCCC	Parallel Concatenated Convolutional Coding
PDNP	Pattern Dependent Noise Prediction
PEG	Progressive Edge Growth
pmf	Probability Mass Function
PMRC	Perpendicular Magnetic Recording Channel

PRML	Partial Response Maximum Likelihood
PR1	Partial Response Class-1
QC-LDPC	Quasi-Cycle Low Density Parity Check
QLDPC	Q-ary Low Density Parity Check
RS	Reed Solomon
RSCC	Recursive Systematic Convolutional Codes
SCCC	Serially Concatenated Convolutional Coding
SER	Sector Error Rate
SISO	Soft-In Soft-Out
SNR	Signal to Noise Ratio
SPC	Single Parity Code
TE	Turbo Equlization
T-EPCC	Tensor EPCC
T-EPCC-RS	Tensor EPCC-RS
T-EPCC-QLDPC	Tensor EPCC-QLDPC
TPC	Turbo Product Code
TPPC	Tensor Product Parity Code
TP-EPCC	Turbo Product EPCC

Chapter 1

Introduction

1.1 Motivation

The advent of high recording density enabling technologies, pioneered by galloping improvements in head and media design and manufacturing processes, has pushed for similar advances in read channel design and error correction coding, driving research efforts into developing channel-capacity-approaching coding schemes based on soft iterative decoding that are also implementation friendly [1, 2]. Soft decodable error correction codes (ECC), mainly, low density parity check (LDPC) codes, would eventually replace conventional Reed-Solomon (RS) outer ECC. RS codes are characterized by simple encoding and decoding schemes. Yet, despite its large minimum distance, a RS code corresponds to a dense parity check matrix that does not lend itself easily to powerful belief propagation (BP) decoding. Hence, although LDPC codes possess lower minimum distances compared to RS for the same codeword length and rate, and thus a higher error floor, iterative decoding of LDPC is a better approach than soft decoding of RS to bridge the gap to channel capacity. Vast literature exist on the various design aspects of LDPC coded systems for magnetic recording applications. These include code construction [3–6], efficient encoding [7,8], decoder optimization [9–11], and performance evaluation [12–14]. In this thesis, we propose

an LDPC based system optimized for the magnetic recording channel that spans contributions in most of these areas.

Another viable direction is to design an inner soft decodable turbo code with outer RS protection, rather than completely replacing RS. This two level coding scheme is backward compatible with the current industry coding paradigm, in which parity check codes along with channel detection post-processing reduce the bit-error rate (BER) at the input of the RS decoder, enhancing the overall system sector error rate (SER) while incurring little added complexity [15–20]. The continued growth in integration scales of silicon devices motivates the introduction of more complex inner parity systems to further improve RS robustness against worsening channel conditions. On the other hand, read channel imperfections including noises, interferences, and defects, are becoming worse as recording densities are grow at an exponential rate. Thus, inner iterative soft decodable turbo systems would serve as an intermediate generation in the ultimate migration towards single-level and two-level LDPC based outer ECC. Along these lines, we devise a turbo equalizer scheme optimized for the recording channel that delivers substantial performance improvements over state of the art turbo code technologies.

Turbo codes [21–24] have been proposed as a practical means to approach the inter-symbol-interference (ISI) channel capacity in what has been termed turbo equalization (TE) [25, 26]. Since then, turbo equalization refers to any soft decodable code that is iteratively decoded by exchanging soft information with the channel matched detector, such as LDPC codes and turbo product codes (TPC) [27, 28]. A standard turbo code is a parallel concatenation of convolutional codes (PCCC) connected by an interleaver, by which the concatenation reduces the probability of low weight error events considerably, improving the overall system bit-error-rate (BER) in the low-to-medium signal-to-noise ratio (SNR) region. A PCCC is decoded by the iterative exchange of soft information between maximum *a posteriori* probability (MAP) decoders matched to the constituent recursive systematic convolutional codes (RSCC). A TE based on the iterative decoding of the PCCC

soft decoder and the channel detector was discussed in [29]. Nonetheless, a simpler serial concatenation of a single RSCC and a precoded channel through an interleaver was found to perform just as well in [30] for wireless communications applications, and in [31] and [32] for magnetic recording applications, and is termed serially concatenated convolutional coding (SCCC). Channel precoding makes the ISI channel appear recursive to the outer interleaved RSCC, which is essential to achieve better turbo gain in the low SNR region (cliff region) [33]. This was first shown in the context of serially concatenated RSCC in AWGN in [34], where it was demonstrated that the inner constituent convolutional code has to be recursive to achieve turbo gain. This was later demonstrated for SCCC-TE based on a single outer RSCC concatenated to the Dicode channel, representing an inner rate-1 code, in [35]. Precoding improves TE gain by enhancing the error weight "spectral thinning" effect, in which the frequency of low-to-medium Euclidean distance errors is uniformly reduced. In this work, we propose an improved spectral thinning approach that aggressively targets the low end of the error Euclidean distance distribution, enhancing BER performance in both the cliff and "error floor" regions. The error floor region is specially important, since an actual TE system would operate there to reduce its sensitivity to SNR variations. The proposed method is based on directly targeting the dominant error patterns of the channel, which are also the lowest Euclidean distance errors, via a matching error correction code, termed the error-pattern correction code (EPCC). The error-pattern correcting code (EPCC) was proposed in [36–38] and was motivated by the well-known observation that the error rate at the channel detector output of an ISI channel is dominated by a few specific known error cluster patterns. This is due to the fact that the channel output energies associated with these error patterns are smaller than those of other patterns, i.e. they have the smallest Euclidean distance from the correct path in the channel detector trellis.

In the proposed TE-EPCC decoder [39], the EPCC MAP decoder works iteratively with the outer RSCC MAP decoder to suppress low Euclidean distance errors at the output of the channel's MAP detector. In particular, we describe a soft-input soft-output (SISO) decoder

for the error-pattern correction code, and consider the incorporation of the SISO decoder as a building block of a turbo-equalization system, with application to perpendicular magnetic recording channels (PMRC). We also derive an upper bound on the BER of TE-EPCC to compare its performance to more conventional precoded and unprecoded TE systems.

The multiparity cyclic EPCC described in [37], concatenated to a RS outer ECC, possesses distinct syndrome sets for all dominant error patterns. To reduce the codes rate penalty, which is a severe SNR degradation in recording applications, a method to increase the code rate was introduced in [38] that also improved EPCC's algebraic single and multiple error-pattern correction capability. In this method, the generator polynomial of the short base EPCC is multiplied by a primitive polynomial that is not already a factor of the generator polynomial. Moreover, the primitive polynomial degree is chosen so as to achieve a certain desired codeword length. Also, [38] describes a Viterbi detection postprocessor that provides error-event reliability information aiding EPCC's syndrome-error mapping to improve its correction accuracy. However, raising the EPCC code rate by extending its codeword length increases the probability of multiple dominant error patterns within the codeword, and this requires increasing the size of the syndrome table considerably to maintain the same correction power, which eventually results in prohibitive decoding complexity.

The recent work described in [40] discusses a more efficient decoding method based on a list decoding strategy, wherein a list of tentative decoder input words are constructed by perturbing the original channel detector output. While the concept of running parallel decoders on such test vectors to find a number of possible codewords is well-known [41, 42], the novel feature of the decoding method presented in [40] lies in the way the test decoder input vectors are constructed. The method of [40] uses reliability measures of local bit patterns in the original channel detector output, rather than the usual bit level reliability measure, to flip bit patterns at different positions to generate likely word errors, with respect to the original channel detector output. The highly accurate and relatively simple single-

pattern correcting decoder then acts on each test word. A number of valid codewords are typically produced at the outputs of the parallel decoders, and the most probable codeword is chosen and released as the final decision. This method provides a considerably improved ability to correct multiple-pattern error events. The decoder is also a soft-input decoder in the sense that the soft information out of the channel detector is utilized.

Nevertheless, the serial concatenation scheme of [40] that proved successful with RS hard decoding and RSCC-based turbo equalization, which we will study here, does not work as well in serial concatenation of long-EPCC and LDPC. The reason is that when LDPC fails, especially in the water-fall region, the sector contains a large number of multiple error occurrences. When many such error events occur in a given EPCC codeword, decoding by any reasonable size list decoder is formidable. Thus, an inner EPCC cannot in any capacity reduce the SER of a serially concatenated outer LDPC. On the other hand, if the EPCC codeword length is decreased substantially, then the number of errors per codeword is reasonable, as long as the overall code rate is somehow kept high. Here, the concept of tensor product construction comes into play.

Tensor product parity codes (TPPC) were first proposed in [43], as the null-space of the parity check matrix resulting from the tensor product of two other parity check matrices corresponding to a variety of code families. As a result, based on the choice of the concatenated codes, TPPC would be classified as an error correction code if constructed from two ECCs, an error detection code (EDC) if constructed from two EDCs, and an error location code (ELC) if constructed from an EDC and an ELC in a special order. In fact, ELCs were introduced earlier in [44] and their algebraic properties studied in detail, but later incorporated in the unified theme of TPPCs in [43]. Furthermore, a generalized class of hard-decodable ELCs was later suggested for application in the magnetic recording channel in [45]. In addition, TPPCs can be generalized by combining a number of codes on various extension fields with shorter binary codes. For this more general case, a decoding algorithm was developed in [46]. An ECC-type TPPC was applied to longitudinal magnetic

recording in [47], and to perpendicular magnetic recording in [48]. In [47], a hard decodable tensor product code based on a single parity code and a BCH code is proposed as an inner code for RS. This code is suitable for low density longitudinal recording channels for which dominant errors have odd weights, such as $\{+\}$ and $\{+,-,+\}$. Also, [47] proposes that the hard decoder passes the corrected parity bits to a Viterbi detector reflecting channel states and parity code states in order to compute the decoder output. Later, [48] presented two methods for combining a tensor-product single parity code with a distance-enhancing constrained code. This later code combination achieved more satisfactory performance with RS as an outer code in high density perpendicular recording channels.

Our goal in this work is to utilize the concept of tensor product concatenation to construct high rate soft-decodable EPCCs on the symbol-level of the outer ECC. The EPCC target error list is matched to the dominant error events normally observed in high density perpendicular recording channels. Since dominant error events in perpendicular recording are not only of odd weight [12], this requires that our EPCC be a multiparity code. However, in this case, a one-shot Viterbi detector matching the channel and parity will have prohibitive complexity. In spite of this, the performance of the optimal decoder of the baseline parity-coded channel can be approached by the low complexity detection postprocessing technique in [40]. We also present in detail a low complexity highly parallel soft decoder for T-EPCC and show that it achieves a better performance-complexity tradeoff compared to more conventional iterative decoding schemes.

1.2 Thesis Outline

The rest of this thesis is organized as follows. In Chapter 2, we consider a turbo-equalizer system that utilizes the error-pattern correcting code as a building component. A soft-input soft-output (SISO) decoder for the error-pattern correcting code (EPCC) is also described. We also introduce a soft-decision generator stage that produces bit-level soft decisions

based on the list of candidate codewords made available at the output of the single-pattern correcting parallel decoders. Deriving the bit-level soft decisions is based on the existing approach of finding probability measures of each candidate codeword, converting them to bit-level reliability measures by grouping the candidate codewords according to the binary value of the given bit position, and then performing group-wise averaging of the codeword probabilities. What is unique in our approach, however, is that we estimate the candidate codeword probability based on the reliability measures of the local patterns in the candidate codeword, rather than on the usual Euclidean distance of the codeword from the detector output. In estimating the reliability of a local pattern, we make use of maximum *a posteriori* probability (MAP)-based, finite-window correlators matched to the dominant error patterns. Furthermore in this chapter, an error weight analysis of EPCC enhanced TE is conducted to arrive at an upper bound that predicts the BER performance advantage of incorporating EPCC at the error floor region. In this area, we will study the error enumerators of the Dicode channel, $1 - D$, in AWGN for two cases: 1) no precoding 2) and precoding with a $1/(1 \oplus D)$ precoder, where we invoke the uniform interleaver assumption so that the channel's input errors after the interleaver are i.i.d and equiprobable. With the utility of these error enumerators, we derive a fully analytic upper bound on the BER of error-pattern correcting turbo codes based on a RSCC. We Also compare the derived upper bound to the upper bound on BER of conventional TE systems for precoded and unprecoded channels.

In Chapter 3, we study the performance of the proposed TE-EPCC in channels dominated by data-dependent media noise, which is the major noise source in PMRC. To that end, we present a modification of the EPCC list decoder that can work in the media noise environment. The modified TE-EPCC is compared to conventional TE, the (i.u.d.)-input channel capacity, and an algebraically constructed quasi-cyclic LDPC (QC-LDPC) that rivals the performance of randomly constructed LDPC for sector-size codeword lengths. An alternative concatenation based on the turbo product concept is also presented, termed TP-EPCC, and its performance is measured and compared to the other coding systems.

In Chapter 4, we present a two-level tensor product concatenation of EPCC and standard RS or LDPC as an alternative system to single-level RS and LDPC systems. We present a review of the tensor product coding paradigm and show how to encode tensor EPCC (T-EPCC) to achieve linear time encodability given that the constituent encoder are also linear time encodable. Moreover, we present the tensor product concatenation of EPCC and RS, T-RS-EPCC, as an example tensor product system in order to demonstrate the algebraic structure and properties of the concept. For example, the hard decoder of T-RS-EPCC would serve as starting point for the development of the soft decoder of T-LDPC-EPCC, where soft multi-level LLR (mlLLR) algebra is a natural extension of the finite field algebra that is employed in the hard decoder. In addition, we develop a novel fully analytic RS SER estimation technique in ISI channels. The proposed method, termed the fully-analytic multinomial method, enables the performance comparison of single-level RS and the constituent RS in the two-level T-RS-EPCC, assuming perfect EPCC operation. The purpose of such experiment, is to find out wether the more sever channel conditions facing the constituent RS will not wipe out its performance leverage, which stems from its significantly lower code rate.

In Chapter 5, we apply the design lessons of Chapter 4 in the design of a novel two-level concatenation of EPCC and Q-ary LDPC (QLDPC), in what we term T-EPCC-QLDPC. In that effort, we explain our choice of the component QLDPC and justify the use of the progressive edge growth (PEG) algorithm with a quasi-cyclic constraint to construct QLDPC. We then propose in detail a soft iterative practical decoder of T-EPCC-QLDPC and evaluate its SER performance with respect to single-level QLDPC and binary LDPC. Finally, Chapter 6 contains the summary of the thesis and suggested future work directions.

Chapter 2

The Error-Pattern Correcting Turbo Equalizer

2.1 Review of the Error-Pattern-Correcting Code

The cyclic codes described in [37] are based on construction of a generator polynomial $g(x)$ that gives rise to distinct syndrome sets for all targeted dominant error patterns. It has been shown that such a $g(x)$ can be obtained from the irreducible factors making up the polynomial representations of the dominant error patterns. The code can be further improved by introducing another factor in $g(x)$, namely, a primitive polynomial that is not already a factor of $g(x)$ [38]. The results are an increased code rate, improved single-error-pattern correction accuracy (via reduced miss-correction probability), and capability to correct some important multiple-pattern events based on a increased number of distinct syndrome patterns.

We start by constructing a cyclic code targeting the set of l_{max} dominant error events

$$\{e_k^{(1)}(x), e_k^{(2)}(x), ..., e_k^{(l_{max})}(x)\}$$

represented as polynomials on $GF(2)$ that can occur at any starting position k in the code-word of length l_T. A syndrome of error $e^{(i)}(x)$ at position k is defined as $s_k^{(i)}(x) =$

$e_k^{(i)}(x)\mathbf{mod}g(x)$, with $g(x)$ being the generator polynomial of the code and $\mathbf{mod}$ the polynomial modulus operation. A syndrome set $\mathbf{S}_i$ for error type $e^{(i)}(x)$ contains elements corresponding to all cyclic shifts of polynomial $e^{(i)}(x)$; elements of $\mathbf{S}_k$ are thus related by $s_k^{(i)} \equiv x^{i-j}s_k^{(j)}\mathbf{mod}g(x)$.

For unambiguous decoding of $e^{(i)}(x)$ and $e^{(j)}(x)$, $\forall\{i,j\}$, we must have $\mathbf{S}_i \cap \mathbf{S}_j = \oslash$. This design requirement constrains $g(x)$ to have distinct greatest common divisors with all $e^{(i)}(x)$. However, even if this constraint is satisfied, an element in $\mathbf{S}_i$ can still map to more than one position, i.e., the period of the syndrome set - and period of $g(x)$ - can be less than l_{max}. Also, this constraint is sufficient but not necessary. As shown in [37], there may exist a lower degree $g(x)$ that can yield distinct syndrome sets for the targeted error polynomials, resulting in a higher rate EPCC. A search method to find this $g(x)$ is already discussed in detail in [37] and [40].

We now describe the construction and properties of some EPCC codes that will be deployed throughout the thesis in the design of different turbo systems based on EPCC. We target the dominant error events of the Dicode channel, PR1 channel, and the equalized monic channel $1+\alpha D,\ \alpha < 1$, which is suitable as a partial response target in perpendicular magnetic recording read channels. For all of these channel responses, the dominant errors are: $e^{(1)}(x) = 1$, $e^{(2)}(x) = 1 + x$, $e^{(3)}(x) = 1 + x + x^2$, etc., i.e. polynomials on $GF(2)$ for which all powers of x have nonzero coefficients.

For the purpose of designing EPCC codes for use in TE-EPCC and TP-EPCC, the component EPCC code rate should be very high. To maintain high rate in the serial concatenation of EPCC and RSCC (TE-EPCC), or in the parallel concatenation of EPCC and an outer block parity code (TP-EPCC), the EPCC codeword has to be extended to a few hundred bits, without proportionally increasing the number of parity bits required to achieve accurate single-error occurrence correction capability. In the following, we construct the following long EPCC codes:

- *Long EPCC 1*: Targeting error polynomials up to degree 9, we get the generator

polynomial $g(x) = 1 + x^3 + x^5 + x^8$ of period 30, via the search procedure in [37]. Choosing a codeword length of 30, 10 distinct, non-overlapping syndrome sets are utilized to distinguish the 10 target errors. However, the resulting $(30, 22)$ EPCC has rate 0.73 which incurs high rate penalty. By multiplying the base EPCC generator polynomial with the primitive polynomial $1 + x + x^6$, which is not a factor of any of the targeted errors, we obtain the extended generator polynomial $g_e(x) = 1 + x + x^3 + x^4 + x^5 + x^8 + x^{11} + x^{14}$, which corresponds to the extended $(630, 616)$ EPCC code of rate 0.98, and 14 parity bits. Then, as shown in [37], syndrome sets $\mathbf{S}_1$, $\mathbf{S}_3$, $\mathbf{S}_7$, and $\mathbf{S}_9$ have period 630 and thus can be decoded without ambiguity. On the other hand, syndrome sets $\mathbf{S}_2$, $\mathbf{S}_4$, $\mathbf{S}_6$, and $\mathbf{S}_8$ have period 315, decoding to one of two positions. The worst would be $\mathbf{S}_5$ of period 126, and $\mathbf{S}_{10}$ of period 63, which decode to 5 and 10 possible positions, respectively. Still, the algebraic decoder can quickly shrink the number of possible error positions to few positions by checking the data support, and then would choose the one position with highest local reliability.

- *Long EPCC 2*: Shorter, lower rate, EPCC codes can be obtained by shortening the $(630, 616)$ EPCC. For example, a $(126, 112)$ EPCC of rate 0.89 can be derived this way with all syndromes sets, excluding syndrome set $\mathbf{S}_{10}$, having period 126, and thus are decodable without ambiguity.

- *Long EPCC 3*: Targeting error polynomials up to degree 9, but excluding $e^{(7)}(x)$, we can extend the base generator polynomial $g(x) = 1 + x^3 + x^5 + x^8$ by multiplying it by the primitive polynomial $1 + x + x^3$, which we could not use before because its a factor of the polynomial representation of $e^{(7)}(x)$. The resulting code is a $(210, 199)$ EPCC of rate 0.95, 11 parity bits, and extended generator polynomial $g_e(x) = 1 + x + x^4 + x^5 + x^9 + x^{11}$. The syndrome set periods of this code are shown in Table 2.1.

- *Long EPCC 4*: Targeting the 4 shortest dominant error patterns, the search procedure produces the base generator polynomial $g(x) = 1 + x + x^3 + x^4$. Extending this

polynomial using the primitive polynomial $1+x^2+x^5$, we get the extended generator polynomial $g_e(x) = 1 + x + x^2 + x^4 + x^8 + x^9$, which corresponds to a $(186, 177)$ EPCC of rate 0.95, and 9 parity bits. The syndrome set periods of this code are shown in Table 2.1.

For the purpose of designing tensor product concatenated codes, the EPCC codeword would be very short, in the order of the outer ECC symbol length. In the following, we construct the following short EPCC codes:

- *Short EPCC 1*: Targeting error polynomials up to degree 4, we get the generator polynomial $g(x) = 1 + x + x^3 + x^5 + x^6$ of period 12. Choosing a codeword length of 12, 5 distinct, non-overlapping syndrome sets are utilized to distinguish the 5 target errors. Then, syndrome set $\mathbf{S}_3$ will have period 6, while all other sets have period 12. A syndrome set of period 6 means that each syndrome decodes to one of 2 possible error positions within the 12-bit codeword. Nonetheless, $e^{(3)}(x)$ can be decoded reliably via channel reliability information and the polarity of data support. The low code rate of 0.5 makes this code unattractive as an inner code in a serial concatenation setup for recording channel applications. However, as we will see later, a tensor code setup makes it practical to use such powerful codes for recording applications.

- *Short EPCC 2*: Targeting error polynomials up to degree 9, we have to record more redundancy. To accomplish this feat, a cyclic code with 8 parity bits, code rate 0.56, and a generator polynomial $g(x) = 1 + x^2 + x^3 + x^5 + x^6 + x^8$ of period 18 is found by the search procedure in [37]. Then, syndrome sets $\mathbf{S}_1$, $\mathbf{S}_3$, $\mathbf{S}_5$, and $\mathbf{S}_7$ have period 18 and thus can be decoded without ambiguity. While syndrome sets $\mathbf{S}_2$, $\mathbf{S}_4$, $\mathbf{S}_6$, and $\mathbf{S}_{10}$ have period 9, decoding to one of two positions. The worst would be $\mathbf{S}_9$ of period 2, which would decode to one of 9 possible positions. Still, the algebraic decoder can quickly shrink this number to few positions by checking the data support, and then would choose the one position with highest local reliability.

Table 2.1: Syndrome set periods of various EPCC codes.

Target error	L-EPCC 1	L-EPCC 2	L-EPCC 3	L-EPCC 4	S-EPCC 1	S-EPCC 2	S-EPCC 3
1	630	126	210	186	12	18	24
$(1+x)$	315	126	105	93	12	9	24
$(1+x+x^2)$	630	126	70	62	6	18	12
$(1+x)^3$	315	126	105	93	12	9	24
$(1+x+x^2+x^3+x^4)$	126	126	42	–	12	18	24
$(1+x)(1+x+x^2)^2$	315	126	35	–	–	9	12
$(1+x+x^3)(1+x^2+x^3)$	630	126	–	–	–	18	24
$(1+x)^7$	315	126	105	–	–	9	24
$(1+x+x^2)(1+x^3+x^6)$	630	126	70	–	–	2	12
$(1+x)(1+x+x^2+x^3+x^4)^2$	63	63	21	–	–	9	24

- *Short EPCC 3*: Targeting error polynomials up to degree 9 again, but seeking a longer codeword, we need to record more redundancy than the second short EPCC. To that end, the search procedure produces the generator polynomial $g(x) = 1 + x + x^2 + x^4 + x^5 + x^6 + x^8 + x^9 + x^{10}$ of period 24. The resulting $(24, 14)$ EPCC of rate 0.58 has the syndrome set periods in Table 2.1.

2.2 A SISO Decoder for EPCC

In our TE setup, the EPCC is matched to the ISI channel and serves as an inner code for an outer interleaved convolutional code. Since the EPCC maintains substantial error correction power while having a high code rate that is close to 1, the hope is that the redistribution of redundancy between the EPCC and the outer code in a TE setup would improve overall system performance. In SISO decoding of EPCC, EPCC utilizes the output extrinsic information coming from the channel detector as *a priori* information in calculating error pattern *a posteriori* probabilities. Then, after generating a list of the most probable candidate codewords, the decoder uses the list to calculate the output bit-level decision reliabilities that serve as the *a priori* information for other SISO elements in the soft iterative system.

The list decoding scheme employed here has been discussed in [40]; in this chapter, we formulate the channel-matched decoder as a SISO decoder and modify it to handle medium-noise environments. The list decoder of [40] is philosophically similar to Chase decoding [41] in the sense of generating test vectors at the parallel decoder input. The difference is that in the set up of [40], during the test word construction stage the EPCC decoder flips multi-bit error patterns, not the individual bits, to form a set of test error words that are decoded by an array of single-pattern correcting decoders. The parallel decoder outputs form the final list of valid candidate codewords.

In summary, the list-decoding/soft-output generation process we consider has three phases:

- The test error word list is generated by inserting the most probable combination of local error patterns to the channel detector output.
- An array of parallel single-pattern correcting decoders decode the test words to produce a list of valid codewords.
- The list of candidate codewords is used to generate bit-level decisions along with their reliabilities.

We now discuss each phase in detail.

2.2.1 Generation of the Test Error Word List

First consider an event that the channel detector output word, which we will call the maximum likelihood (ML) word (assuming the channel detector outputs the ML word or its close approximation), is corrupted by a dominant pattern $e^{(i)}(x)$ starting at bit location k_i. We can think of a local error pattern (with respect to the ML word) denoted by $x^{k_i}e^{(i)}(x)$ of type i and starting location k_i. A test word may contain one or more such local error patterns. We wish to produce a number of highly probable test words. The probability measure of a given test word with a particular combination of local patterns can be determined by estimating the probability of dominant patterns at a given starting position. These local probability measures are calculated via the local correlator. Specifically, those type/location pairs that are most probable in the sense of maximizing the correlator function are used to construct test words. The requirement to have M-error-pattern-correction capability using the single-pattern correcting decoders, dictates that test words must include up to $M-1$ local error patterns. Starting from the m most probable such local error patterns $x^{k_i}e^{(i)}(x)$'s (i.e., corresponding to the m most probable pairings of (i, k_i)), one can think of $\binom{m}{1} + \binom{m}{2} + ... + \binom{m}{M-1}$ ways of corrupting the ML word with up to $M-1$ local error patterns. From this large set of potential combinations, a relatively small subset of most probable combinations needs to be chosen to maintain reasonable complexity. One

can think of many different ways of effectively constructing such a list [40], based on the probable local error patterns that have been identified.

2.2.2 Parallel Algebraic Decoding

The list of test error words generated above is delivered to an array of single-error-pattern correcting decoders that work in parallel to generate the candidate codeword list. The number of parallel decoders is identical to the size of the test word list, and is a crucial parameter that controls the EPCC decoder's complexity/performance tradeoff. In the decoding process, each decoder searches the space in the proximity of its input test word for valid codewords with zero syndrome. Since it is a single-pattern correcting code, the space of interest around the test word is the union of the neighborhoods centered around the test word each having radius chosen solely from the list of most probable local error patterns. In the event of finding a valid codeword in the search space, the decoder releases it as its decoded output. Otherwise, with no codeword found, the decoder does not contribute to the final list. As such, the size of the generated final list is variable.

2.2.3 Generation of Soft Output

The candidate codeword list constructed by our "pattern-level" list decoder is used to evaluate the more familiar bit-level reliabilities that constitute the output soft information supplied by the EPCC SISO decoder. Typically, the candidate codeword list forms a reduced space for the maximum likelihood decoder (MLD) to search for the maximum-likelihood codeword given the observed word. When the observed word is the output of an additive white gaussian noise (AWGN) channel, the metric that specifies the likelihood of the codeword is the Euclidean distance to the observed word. In our list soft-decoder formulation, the resolution of the space of possible codewords is measured in units of "error-patterns" rather than error bits. As a result, we measure the probability of a candidate codeword given the observed word by the product of the probabilities of each "local" error pattern

forming the candidate word. Specifically, let $\mathbf{c}$ represent a candidate codeword with, say, K error-pattern corruption with respect to the ML word $\hat{\mathbf{c}}$. Then, the *a posteriori* probability of this particular test word, $Pr(\mathbf{c}/\hat{\mathbf{c}}, \mathbf{r})$, is estimated by multiplying the probability estimates of the K local patterns, given the channel observation $\mathbf{r}$ at the detector input. The probability of $e^{(i)}(x)$ starting at bit location k_i, i.e., the local error pattern $x^{k_i}e^{(i)}(x)$, can be computed via (3.8) to be discussed in the next chapter. Also, the actual probability computation is done in the log-domain using the metrics (3.13) for each constituent error-pattern. The probability measures of candidate codewords are further differentiated apart by the utility of the "pattern-level" *a priori* probability term in (3.13), which is available by grouping the constituent "bit-level" *a priori* probabilities of the bits forming the error patterns, as reflected by (3.11) in the next chapter, for each constituent error pattern of the candidate word.

Given the list of codewords and their accompanying *a posteriori* probabilities, the reliability λ_k of the coded bit c_k is evaluated as, see Figure 2.1:

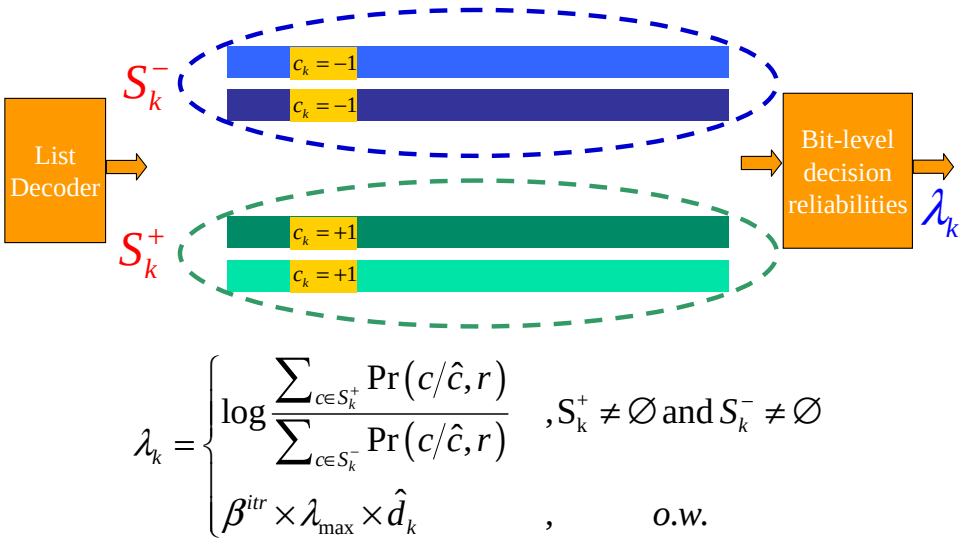
$$\lambda_k = \log \frac{\sum_{\mathbf{c}\in\mathbf{S}_k^+} Pr(\mathbf{c}/\hat{\mathbf{c}}, \mathbf{r})}{\sum_{\mathbf{c}\in\mathbf{S}_k^-} Pr(\mathbf{c}/\hat{\mathbf{c}}, \mathbf{r})} \tag{2.1}$$

where $\mathbf{S}_k^+$ is the set of candidate codewords where $c_k = +1$, and $\mathbf{S}_k^-$ is the set of candidate codewords where $c_k = -1$. The quantity in (2.1) is utilized when the candidate codewords do not all agree on the bit decision for location k. In the event that all codewords do agree on the decision for c_k, a method used by [49] is adopted for generating soft information as follows

$$\lambda_k = \beta^{iter} \times \lambda_{max} \times \hat{d}_k \tag{2.2}$$

where $\hat{d}_k$ is the bipolar representation of the agreed-upon decision, λ_{max} is a preset value for the maximum reliability at convergence of turbo performance, and the multiplier $\beta^{iter} < 1$ will prove useful when incorporating the EPCC SISO decoder in an iterative system. Note that in an iterative system the level of confidence in bit decisions is lower at the initial iterations, and thus multiplying the generated log likelihood ratios by the back-off factor

β^{iter} reduces the risk of error propagation. It will also be necessary to use (2.2) to generate soft information if the reliability threshold check activates only one algebraic single error pattern correction decoder rather than the list decoder, as discussed in the next section.



$$\lambda_k = \begin{cases} \log \dfrac{\sum_{c \in S_k^+} \Pr(c/\hat{c}, r)}{\sum_{c \in S_k^-} \Pr(c/\hat{c}, r)} & , \mathrm{S}_\mathrm{k}^+ \neq \varnothing \text{ and } S_k^- \neq \varnothing \\ \beta^{itr} \times \lambda_{\max} \times \hat{d}_k & , \quad o.w. \end{cases}$$

Figure 2.1: Generation of bit-level LLR in TE-EPCC

2.3 A TE Incorporating EPCC SISO Decoder

The proposed EPCC SISO decoder is now ready to be used as a building block in turbo systems. Since the EPCC is matched to the ISI channel, no interleaving should be present between EPCC and the channel. On the other hand, an interleaver is essential between the EPCC and the outer recursive systematic convolutional code (RSCC). A legitimate question to be posed is whether the EPCC and TE would benefit from working together. A major step in answering this question is the following observation. We see that the set of most dominant error patterns produced by an ISI channel is not a sensitive function of its operating signal-to-noise ratio (SNR). This means that iteratively-improved *a priori* information fed to the channel detector, which can be viewed as providing a boost in SNR, would still give the same set of dominant error patterns that can be further corrected by the

EPCC, resulting eventually in an improved *a priori* input to the channel detector and the same dominant error set, provided no serious miss-correction is present, and so on. This is a strong motivation for the integration of the two systems, since then, gradual iterative improvement would be achieved. This behavior is in contrast to the performance saturation phenomena in a conventional turbo system within a few channel iterations. The proposed setup is shown in Figure 2.2. The Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [50], which incorporating pattern-dependent noise-prediction (PDNP) [51] [52] in media noise dominated environments, generates extrinsic log-likelihood ratios (LLR) λ_k^e that can be used by the EPCC along with channel observation r_k to estimate the probability of single error patterns. Using the list of candidate codewords the EPCC SISO decoder calculates extrinsic bit-level reliabilities $\tilde{\lambda}_k^e$ that are fed as channel observations to the RSCC decoder after deinterleaving. The RSCC BCJR-based decoder in turn generates its own extrinsic information that is fed back to the BCJR/PDNP detector after interleaving. In the EPCC decoder, a decision is first made on whether or not the decoder input contains a single error pattern via the syndrome check. If the initial syndrome check indicates either an error free input, or else, a single error pattern with the reliability threshold qualification also satisfied [40], then, the corrected output is released accordingly, and (2.2) is used to generate the decision reliabilities for each of the hard bits making up the corrected codeword. If not, the list decoder is activated that involves computing correlator-based reliability estimates for local patterns in the ML word, and generation of decision reliabilities using (2.1). Simulations show that the aforementioned strategy of moving between list-decoding and algebraic single pattern decoding results in improved performance compared to running list-decoding all the time, since at later turbo iterations single error-pattern occurrences are more likely, and syndrome-decoding is more robust in such scenarios. One drawback of the TE-EPCC setup involving the serial concatenation of an interleaved RSCC and EPCC is that the parity bits introduced by the addition of the innermost EPCC are not protected by the outer RSCC. Thus, the *a priori* information of these bits are not updated by turbo

channel iterations. Although error events in those bits do not count towards the final bit error rate, since EPCC parity bits get discarded prior to deinterleaving, they can still pair up with error events elsewhere in the codeword to form multiple-error-pattern occurrences that are not always resolvable by the EPCC-turbo action. One solution around this problem would be to design a turbo product code (TPC) in which the EPCC is the row encoder, in which case its parity bits are protected by the column encoder. An interesting design problem for TPC there would be to ensure that the column encoder does not break up dominant error-patterns into unrecognized error events that are not targeted by EPCC of the next inner TPC iteration.

2.4 Error-Rate Analysis of TE-EPCC

2.4.1 Weight Enumeration of Recursive Convolutional Codes

For the purpose of bounding the BER of turbo equalizers we need to approximate the input and output codeword weight distributions of convolutional codes. Specifically, we need to find the Hamming weight enumerator $\mathbf{A}(d,i)$, which represents the number of codeword sequences of weight d that originated from weight i information sequences. From $\mathbf{A}(d,i)$ we can obtain the marginal enumerators:

$$\mathbf{A}(d) = \sum_i \mathbf{A}(d,i) \tag{2.3}$$

which represents the number of codeword sequences of weight d, and

$$\bar{\mathbf{A}}(d) = \frac{\sum_i i\mathbf{A}(d,i)}{\mathbf{A}(d)} \tag{2.4}$$

which represents the average input Hamming weight of codewords of weight d. To find the weight enumerators, we utilize the state transition matrix $\mathbf{M}(I,D)$ of the recursive convolutional code derived from its time non-varying trellis. The element (i,j) of a state transition matrix of a convolutional code corresponds to the parameters of the transition

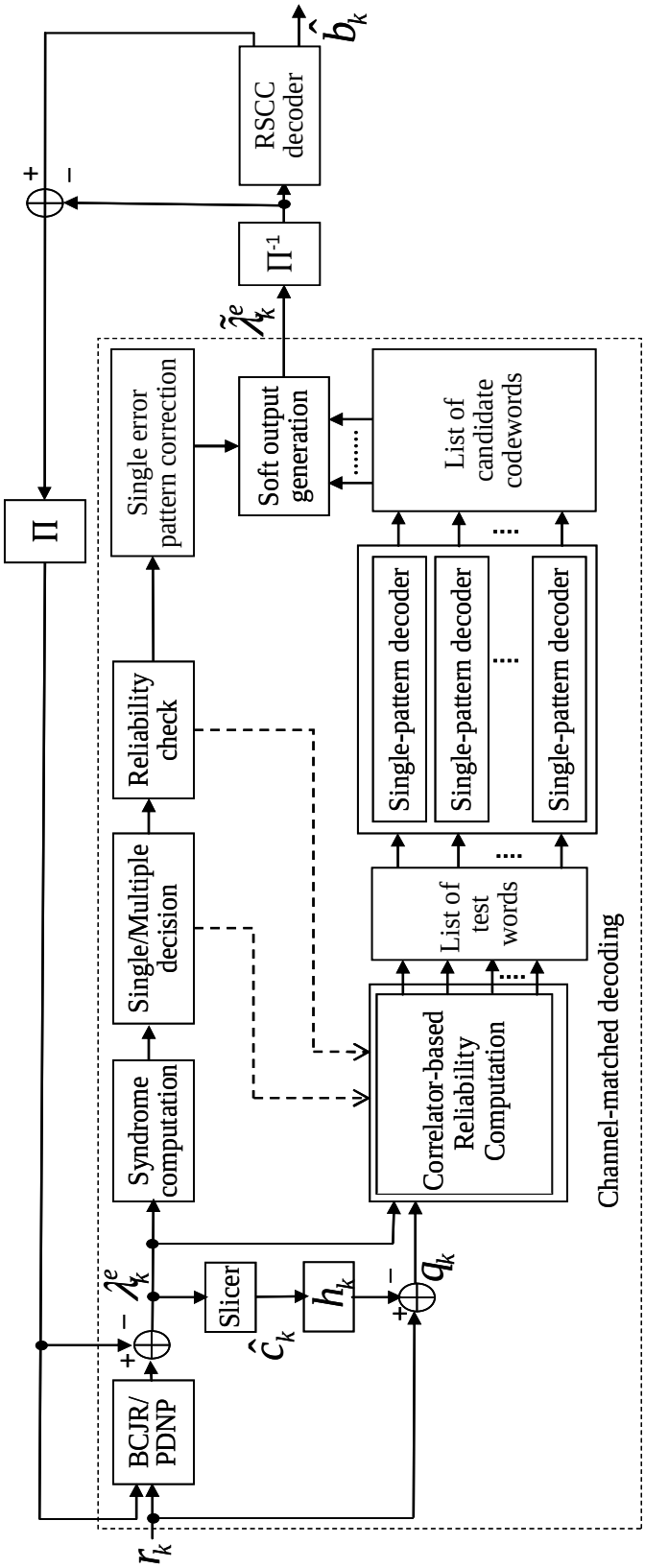


Figure 2.2: TE-EPCC block diagram

from state s_i to state s_j in the trellis representation. In particular, a state transition from state s_i to state s_j initiated by input bit $b_i \in \{0,1\}$ and resulting in output bit $b_o \in \{0,1\}$ corresponds to the transition matrix monomial element $\mathbf{M}_{i,j} = I^{b_i} D^{b_o}$, where the variable power of variable I accumulates the growth in input weight per trellis transition, while the power of variable D accumulates output weight. However, if a state s_j is unreachable from state s_i, then $\mathbf{M}_{i,j} = 0$. Without loss of generality, we define a codeword as a valid path in the trellis that starts and ends in state 0. Then, the weight enumerator of a sequence of $L + \nu$ coded bits corresponding to a sequence of $L + \nu$ state transitions is found from element $(0,0)$ of the matrix $\mathbf{M}(I,D)^L \mathbf{M}(1,D)^\nu$, where ν corresponds to the constrained length of the code and $\mathbf{M}(1,D)^\nu$ is associated with the transmitted tail-bits, for which, no input weight is accumulated. We represent element $(0,0)$ of $\mathbf{M}(I,D)^L \mathbf{M}(1,D)^\nu$ by the multinomial:

$$\mathbf{T}(I, D, L+\nu) = \sum_{i \geq 0} \sum_{d \geq 0} t(i,d) I^i D^d, \tag{2.5}$$

where the coefficient of monomial $I^i D^d$ provides $\mathbf{A}(d,i)$ for a codeword of length $L + \nu$ starting and ending at state 0, and we call this multinomial the weight transfer function of the codeword. In this work, we are interested in systematic convolutional recursive codes of rate $\frac{1}{2}$ punctured to rate $\frac{n-1}{n}$, where $n \geq 3$ is an integer. Since the systematic base code has rate $\frac{1}{2}$, we modify the elements of $\mathbf{M}$ corresponding to a valid transition to $\mathbf{M}_{i,j} = I^{b_i} D^{b_o} D^{b_i}$. Furthermore, we only puncture parity bits, and as such, punctured trellis sections have valid transitions corresponding to $\mathbf{M}^{(p)}_{i,j} = I^{b_i} D^{b_i}$. Hence, assuming a codeword length N that is an integer multiple of the puncturing period n, we first find the transfer function of one puncturing period n as the $(0,0)$ element of:

$$\mathbf{B}(I,D) = \left[\mathbf{M}(I,D)^{(p)}\right]^{n-1} \mathbf{M}(I,D). \tag{2.6}$$

Then, the weight transfer function of the codeword is:

$$\mathbf{T}(I, D, N+\nu) = [\mathbf{B}^{N/n}(I,D)\mathbf{M}(1,D)^\nu]_{0,0}. \tag{2.7}$$

An efficient method to recursively compute $\mathbf{T}$ was demonstrated in [53]. The method is based on the fact that we only require the $(0,0)$ element in the matrix product of (2.7) to obtain the weight transfer function. Then, if we define b^T to be the first raw of $\mathbf{B}$, and observe that $b^T\mathbf{B}$ is just the first row of $\mathbf{B}^2$, we can thus apply the equation $b^T(k) = b^T(k-1)\mathbf{B}$ recursively $N/n-1$ times to find the first row of $\mathbf{B}^{N/n}$. In addition, starting from $b^T(N/n-1)$ we apply $b^T(k) = b^T(k-1)\mathbf{B}$ another ν times to find the first row of $\mathbf{B}^{N/n}(I,D)\mathbf{M}(1,D)^{\nu}$, for which the $(0,0)$ element is the weight transfer function of the codeword of length $N+\nu$. Tables 2.3 and 2.2 list the output and average input weight enumerators, respectively, for puncturing rates $\frac{2}{3}$, $\frac{3}{4}$, $\frac{4}{5}$, $\frac{5}{6}$, $\frac{6}{7}$, $\frac{7}{8}$, and $\frac{8}{9}$ for a codeword of length $1K$ bits, feed-forward generator polynomial 5 in octal format, feed-backward generator polynomial 7, and constrained length 2. We use the representation (g_0, g_1) to represent the connections of the RSCC, where g_0 is the feed-backward generator polynomial and g_1 is the feed-forward generator polynomial. For the record, this convolutional code has the trellis branch weight enumerators shown in Figure 2.3, and the transition matrix:

$$\mathbf{M}(I,D) = \begin{pmatrix} 1 & 0 & ID^2 & 0 \\ ID^2 & 0 & 1 & 0 \\ 0 & ID & 0 & D \\ 0 & D & 0 & ID \end{pmatrix} \tag{2.8}$$

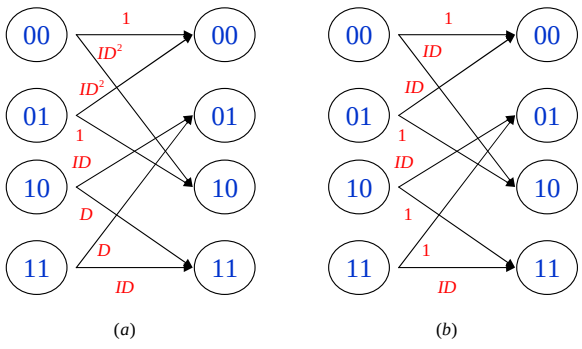


Figure 2.3: (a) Trellis section for the $(7,5)$ RSCC, (b) punctured trellis section.

Table 2.2: Average input Hamming weight enumerators, $\bar{\mathbf{A}}(d_H)$, of different punctured rate RSCC with $g_0 = 7_8$, and $g_1 = 5_8$.

d_H	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{4}{5}$	$\frac{5}{6}$	$\frac{6}{7}$	$\frac{7}{8}$	$\frac{8}{9}$
2	0	0	2	2	2	2	2
3	2.9969	2.3993	2.5984	2.6143	2.789	2.7209	2.7218
4	2.4985	2.996	3.7186	3.8119	3.9612	3.9333	3.9356
5	3.1399	3.9889	4.5433	4.5756	4.7588	4.7002	4.7028
6	5.5550	4.5921	5.3291	5.4561	5.8732	5.7537	5.7594
7	5.4027	5.3433	6.2744	6.389	6.7124	6.6252	6.6318
8	5.6892	6.0505	7.1252	7.2418	7.722	7.5733	7.584

2.4.2 Upper Bound on BER of TE-EPCC

Approximate Bound on the ML BER

We follow a similar approach in bounding the BER of TE-EPCC to the the one proposed in [35] and [54] to bound the BER of conventional TE. Although this method is tailored to Dicode channels, $1 - D$ response, and PR1 channels, $1 + D$ response, it is general enough to demonstrate the tradeoffs and design constraints involved in the design of turbo codes for more general channels. In the proposed approach, we show how the BER is function of the error Euclidean distance distribution of the overall system. Then, we argue for TE-EPCC enhanced performance by the virtue of its reduction of occurrence frequencies of low Euclidean distances in the overall distance distribution, where it will also be shown that these low Euclidean distance components of the distribution dominate the system BER. Following the notation of [35], the maximum likelihood (ML) union bound on word error rate of

Table 2.3: Output Hamming weight enumerators, $\mathbf{A}(d_H)$, of different punctured rate RSCC with $g_0 = 7_8$, and $g_1 = 5_8$.

d_H	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{4}{5}$	$\frac{5}{6}$	$\frac{6}{7}$	$\frac{7}{8}$	$\frac{8}{9}$
2	0	0	196	328	10293	866	997
3	328	1468	2929	4244	102354	8855	11111
4	1305	122731	31154	74394	19667646	422124	564863
5	4548	482116	614985	1474558	382729192	7828743	11352961
6	66240	2379412	7972230	21844149	19168782758	188004023	294265660
7	460188	142771292	99123766	342302076	483275905780	3866470872	6531165491
8	2416261	3195867262	1336834853	5161705491	14004896473555	77259740792	141284713838

a block code of codebook size M, of equally likely codewords, on an AWGN channel of zero mean and variance σ^2 is

$$P_W \leq \frac{1}{M} \sum_{m=1}^{M} \sum_{\acute{m} \neq m} Q\left(\frac{\parallel \mathbf{x}_m - \mathbf{x}_{\acute{m}} \parallel}{2\sigma}\right) \tag{2.9}$$

where m and $\acute{m}$ are codewords separated by the Euclidean distance $\parallel \mathbf{x}_m - \mathbf{x}_{\acute{m}} \parallel$, and $\mathbf{x}_m$ is the noiseless channel output of m. If there are T_{m,d_E} codewords $\acute{m}$ of noiseless channel output $\mathbf{x}_{\acute{m}}$ at Euclidean distance d_E^2 from $\mathbf{x}_m$, then, we can write (2.9) as follows:

$$\begin{aligned} P_W &\leq \frac{1}{M} \sum_{m=1}^{M} \sum_{d_E=1}^{\infty} T_{m,d_E} Q\left(\frac{d_E}{2\sigma}\right) \\ &= \sum_{d_E=d_{min}}^{\infty} \overline{T}(d_E) Q\left(\frac{d_E}{2\sigma}\right) \end{aligned} \tag{2.10}$$

where $\overline{T}(d_E)$ is the average number of codeword sequences of channel noiseless outputs separated by d_E^2. The associated BER can be shown to be

$$P_b \leq \sum_{d_E=d_{min}}^{\infty} \frac{\overline{T}(d_E)\overline{w}(d_E)}{K} Q\left(\frac{d_E}{2\sigma}\right) \tag{2.11}$$

where K is the number of information bits per codeword sequence, and $\overline{w}(d_E)$ denotes the average Hamming distance between information words that generate codewords m and $\acute{m}$ for which $\mathbf{x}_m$ and $\mathbf{x}_{\acute{m}}$ are separated by d_E^2. We next show how $\overline{T}(d_E)$ is fully characterized by the outer code Hamming weight enumerator $\mathbf{A}(d)$ and the error event characteristics of the channel.

Error Event Analysis of the Dicode Channel

A trellis section of the Dicode channel with no precoding is shown in Figure 2.4. Following the same notation as in [35], the branch label c_i/x_i signifies the coded input bit to the channel, and the corresponding channel output, respectively. Any error word $\mathbf{f}$ with Hamming weight $d = d_H(\mathbf{f})$, can be uniquely decomposed into a concatenation of disjoint error patterns $\mathbf{f}_j$, $j = 1, \ldots, m$, where the index j signifies the order of occurrence of the error

pattern in the codeword. Error patterns $\mathbf{f}_j$ correspond to simple closed error events on the trellis that diverge from and remerge into the correct path without sharing any of the states in between. However, two scenarios can occur when $j = m$, either $\mathbf{f}_m$ remerges with the correct path (closed $\mathbf{f}_m$), or the boundary of the codeword is reached while the two paths are still diverged (open $\mathbf{f}_m$). Nevertheless, we have found in this study that such boundary open error events have no effect on the BER down to 10^{-12}, and since this is well below the BER of interest here, we choose to account only for closed error events in bounding the BER. Still, we will revisit the contribution of boundary error patterns, i.e. open $\mathbf{f}_m$, and quantify its contribution to the Euclidean distance distribution later in the chapter.

In the Dicode channel trellis, diverging and remerging branches result in a Euclidean distance separation of 1 each. Moreover, crossing branches accumulate a Euclidean distance separation of 4, while parallel branches do not contribute to the error Euclidean distance. Hence, two error pattern classes are distinguishable depending on their accumulate Euclidean distance. The first class, (b) in Figure 2.4, has a Euclidean distance $d_E^2 = 2$ irrespective of the error length, and here we emphasize again that we only consider closed error events. This class is denoted by χ^{dom} and is called "the dominant error class", for which all branches, other than the diverging and remerging branches, are parallel. The dominant error class accounts for most of the channel bit errors due to the low Euclidean distance between the correct and erroneous paths. On the other hand, the second class, (c) in Figure 2.4, has both parallel and crossing branching, with its members possessing Euclidian distance $d_E^2 = 2 + 4 \times b_{cr}$, where b_{cr} is the number of crossing branches. The second class contributes much less to the overall system BER, and thus we call it "the non-dominant error class", which is denoted by $\widetilde{\chi}^{dom}$. By the same argument line, the same two classes are distinguishable for the PR1 channel having the trellis shown in Figure 2.5. The only difference is that error events with all crossing branches now generate the class χ^{dom} in Figure 2.5(c).

We design an error-pattern correcting code (EPCC) capable of correcting error code-

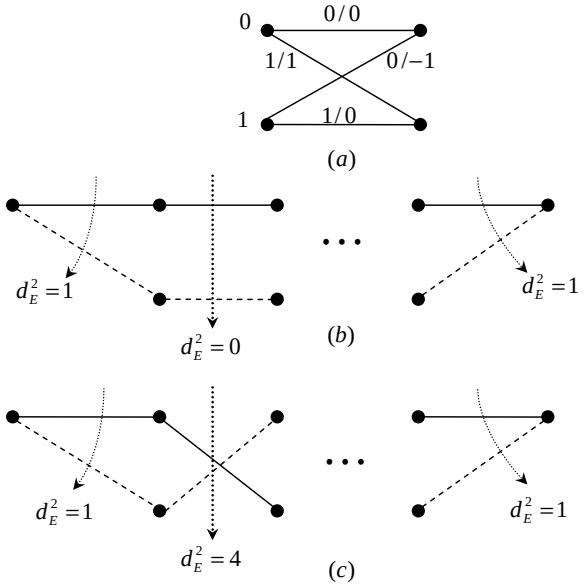


Figure 2.4: (a) Trellis section for the unprecoded Dicode channel $(1 - D)$, (b) dominant error patterns, (c) non-dominant error patterns.

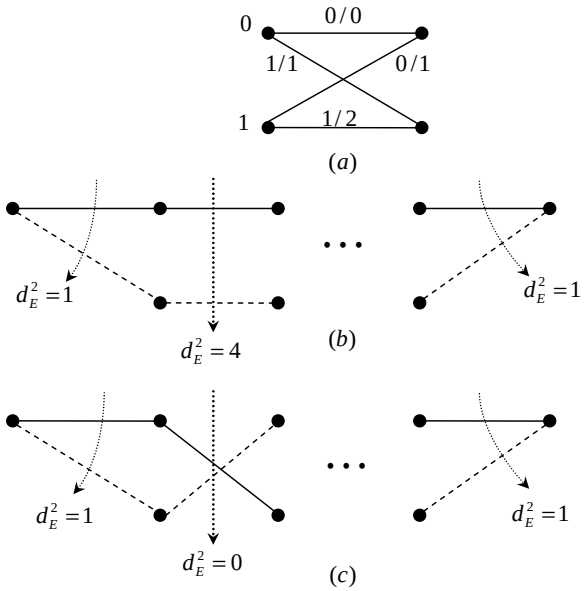


Figure 2.5: (a) Trellis section for the unprecoded PR1 channel $(1 + D)$, (b) non-dominant error patterns, (c) dominant error patterns.

words $\mathbf{f}$ that are decomposable into disjoint error patterns $\mathbf{f}_j$ that all belong to the dominant error class, i.e. $\mathbf{f}_j \in \chi^{dom}, \forall j$. In order to evaluate the BER performance of EPCC we need to find the new Euclidean distance distribution modified by EPCC. However, it would be easier to first find the Euclidean distance distribution before EPCC correction is turned on. We assume throughout that code bit values are i.i.d and equiprobable, which is a valid assumption for high rate codes. Suppose an error word $\mathbf{f}$, of Hamming weight $d_H(\mathbf{f}) = d$, is composed of m_{dom} error patterns $\mathbf{f}_j \in \chi^{dom}$, and $\widetilde{m}_{dom} = m - m_{dom}$ error patterns $\mathbf{f}_j \in \widetilde{\chi}^{dom}$. A dominant error pattern $\mathbf{f}_j$ of length $l_j = d_H(\mathbf{f}_j)$ will have probability $\left(\frac{1}{2}\right)^{l_j-1}$. On the other hand, a non-dominant error pattern $\mathbf{f}_j$ of length l_j and b_{cr} crossing branches will have a probability of $\begin{pmatrix} l_j - 1 \\ b_{cr} \end{pmatrix} \left(\frac{1}{2}\right)^{l_j-1}$. Therefore, the probability distribution of d_E^2 is given by:

$$\Pr(d_E|\, d, m) = \begin{cases} \begin{pmatrix} d-m \\ \frac{d_E^2-2m}{4} \end{pmatrix} \left(\frac{1}{2}\right)^{d-m}, & \frac{d_E^2-2m}{4} > 0 \text{ integer},\ m_{dom} < m \\ \left(\frac{1}{2}\right)^{d-m_{dom}}, & d_E^2 = 2m_{dom},\ m_{dom} = m \\ 0, & \text{otherwise} \end{cases} \tag{2.12}$$

which is the conditional probability of of an error word of Euclidean distance d_E^2, given that its Hamming weight is d, and has m multiple error pattern occurrences, of which m_{dom} belong to χ^{dom}.

Error Euclidean Distance Distribution of TE-EPCC

We develop a method to construct the error Euclidean distance distribution of TE-EPCC, for which the comparable distance distribution of TE is a special case when EPCC is turned off. Consider a serial concatenation of EPCC and an interleaved recursive systematic convolutional code (RSCC) of length N. There are L EPCC subcodes in each interleave, each of length $N_c = \frac{N}{L}$, where EPCC can correct up to m_c multiple occurrences per subcode provided that that they all belong to the target set of correctable errors. The target set is

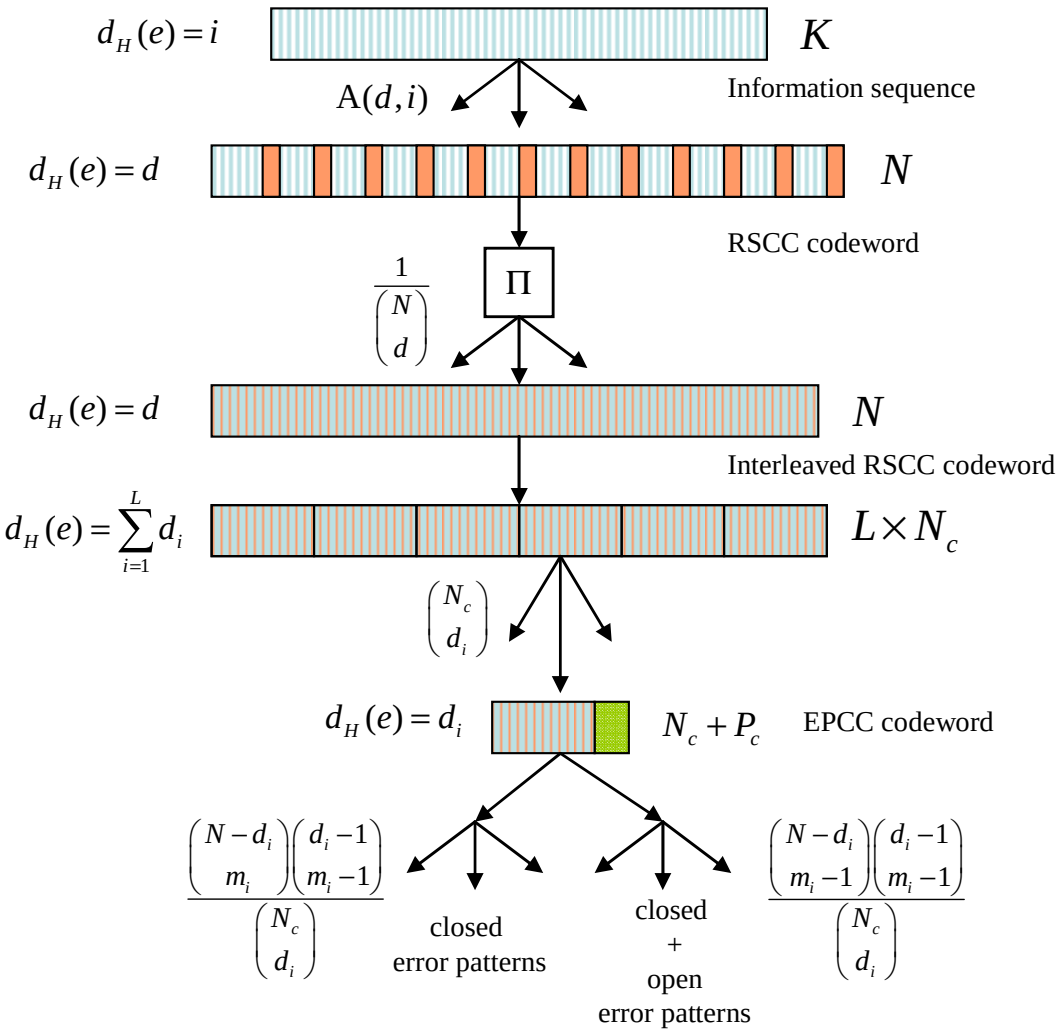


Figure 2.6: Sketch of the method to derive $\overline{T}(d_E)$.

$\{\mathbf{f}_j : \mathbf{f}_j \in \chi^{dom},\ d_H(\mathbf{f}_j) \leq d_c\}$, where d_c is the maximum length correctable error from χ^{dom}. An error in the RSCC codeword of hamming weight d is mapped by the uniform interleaver into all possible $\binom{N}{d}$ interleaved error words $\mathbf{f}$ with equal probability. The interleaved error word is then divided into L EPCC subcodes, each receiving error word $\mathbf{f}^{(i)},\ i = 1, ..., L$ of weights $d_1,\ d_2,\ ...,\ d_L$. Each EPCC error word $\mathbf{f}^{(i)}$ of weight d_i is decomposed into m_i multiple disjoint error pattern occurrences. In the previous section, we found the conditional probability $P(d_E|\ d, m)$ given the error Hamming weight and num-

ber of multiple errors m for a single subcode interleave. To derive the Euclidean distance distribution for a codeword divided into L subcodes, we also are required to evaluate the conditional probability of the decompositions m_i given the EPCC subcode weights d_i. The conditional Euclidean distance probability distribution can be expanded as follows:

$$\begin{aligned} \text{Pr}(d_E|\ d) &= \text{Pr}(d_E|\ d, d_1, ..., d_L) \times \text{Pr}(d_1, ..., d_L|\ d) \\ &= \text{Pr}(d_E|\ d, d_1, ..., d_L, m, m_1, ..., m_L) \times \\ &\quad \text{Pr}(m, m_1, ..., m_L|\ d_1, ..., d_L, d) \times \text{Pr}(d_1, ..., d_L|\ d). \end{aligned} \tag{2.13}$$

Since errors in the L EPCC subcodes are disjoint, (2.13) becomes:

$$\text{Pr}(d_E|\ d) = \sum_{\substack{d_1=0 \\ d=\sum_{i=1}^{L} d_i}}^{d} ... \sum_{d_L=0}^{d} \text{Pr}(d_1, ..., d_L|\ d) \sum_{\substack{m=1 \\ m=\sum_{i=1}^{L} m_i}}^{d} \sum_{m_1=0}^{d_1} ... \sum_{m_L=0}^{d_L} \text{Pr}(d_E|\ d, m) \prod_{i=1}^{L} \text{Pr}(m_i|\ d_i) \tag{2.14}$$

The joint conditional probability $\text{Pr}(d_1, ..., d_L|\ d)$ in (2.14) is the probability of dividing the $\begin{pmatrix} N \\ d \end{pmatrix}$ possible instants of the interleaved error word $\mathbf{f}$, of Hamming weight d, into the error word sequence $\mathbf{f}^{(i)}$ with associated Hamming weight sequence d_i, and is given by

$$\text{Pr}(d_1, ..., d_L|\ d) = \frac{\begin{pmatrix} N_c \\ d_1 \end{pmatrix} \times \begin{pmatrix} N_c \\ d_2 \end{pmatrix} ... \times \begin{pmatrix} N_c \\ d_L \end{pmatrix}}{\begin{pmatrix} N \\ d \end{pmatrix}}. \tag{2.15}$$

Given that there are d_i errors in EPCC subcode i, then, there exists $\begin{pmatrix} d_i - 1 \\ m_i - 1 \end{pmatrix}$ ways by which the Hamming weight d_i error is decomposed into m_i multiple error pattern occurrences, each of length at least 1. Two disjoint error occurrences should be separated at least by the error free distance of the channel, which equals 1 for the Dicode and PR1 channels.

Hence, there are $\begin{pmatrix} N_c - d_i \\ m_i \end{pmatrix}$ ways by which the m_i error patterns, composing $\mathbf{f}^{(i)}$, can be arranged in a subcode i. This is claimed assuming there are no boundary errors (open errors) in the subcode. Given that there are $\begin{pmatrix} N_c \\ d_i \end{pmatrix}$ possible error words $\mathbf{f}^{(i)}$, we get:

$$\Pr(m_i|\, d_i) = \frac{\begin{pmatrix} N_c - d_i \\ m_i \end{pmatrix} \times \begin{pmatrix} d_i - 1 \\ m_i - 1 \end{pmatrix}}{\begin{pmatrix} N_c \\ d_i \end{pmatrix}}. \tag{2.16}$$

A pictorial depiction of the derivation method explained above is shown in Figure 2.6. Substituting (2.12), (2.16), and (2.15) into (2.14), we get an expression for the distribution of error Euclidean distances while EPCC is turned off as:

$$\begin{aligned} \Pr(d_E|\, d) \;=\; & \frac{1}{\begin{pmatrix} N \\ d \end{pmatrix}} \sum_{\substack{d_1=0 \\ d=\sum_{i=1}^{L} d_i}}^{d} \cdots \sum_{d_L=0}^{d} \sum_{m=1}^{d} \sum_{\substack{m_1=0 \\ m:\, d_E^2-2m=0 \bmod 4,\; m=\sum_{i=1}^{L} m_i}}^{d_1} \cdots \sum_{m_L=0}^{d_L} \\ & \times \begin{pmatrix} d-m \\ \frac{d_E^2-2m}{4} \end{pmatrix} \left(\frac{1}{2}\right)^{d-m} \prod_{j=1}^{L} \begin{pmatrix} N_c - d_j \\ m_j \end{pmatrix} \begin{pmatrix} d_j - 1 \\ m_j - 1 \end{pmatrix} \end{aligned} \tag{2.17}$$

where we define $\begin{pmatrix} 0 \\ 0 \end{pmatrix} = 1$. In addition, the Euclidean distance distribution can be decomposed into two components; a component $\Pr(d_E|\, d, \mathcal{C})$ associated with error words that are correctable by the L EPCC subcodes, and the complimentary component $\Pr(d_E|\, d, \widetilde{\mathcal{C}})$ associated with non-correctable error words. In this case, the Euclidean distance probability distribution of non-correctable error words escaping TE-EPCC is given by

$$\Pr(d_E|\, d, \widetilde{\mathcal{C}}) = \Pr(d_E|\, d) - \Pr(d_E|\, d, \mathcal{C}) \tag{2.18}$$

While the correctable component is given by:

$$\Pr(d_E|\ d,\mathcal{C}) = \frac{1}{\begin{pmatrix} N \\ d \end{pmatrix}} \sum_{\substack{d_1=0 \\ d=\sum_{i=1}^{L} d_i}}^{\min(d,d_c)} \cdots \sum_{d_L=0}^{\min(d,d_c)} \sum_{m=1}^{d} \sum_{\substack{m_1=0 \\ m:\ m=d_E^2/2,\ m=\sum_{i=1}^{L} m_i}}^{\min(d_1,m_c)} \cdots \sum_{m_L=0}^{\min(d_L,m_c)} \times \left(\frac{1}{2}\right)^{d-m} \prod_{j=1}^{L} \begin{pmatrix} N_c - d_j \\ m_j \end{pmatrix} \begin{pmatrix} d_j - 1 \\ m_j - 1 \end{pmatrix} \tag{2.19}$$

where for the sake of simplicity in the derivation of (2.19), we assumed that an EPCC subcode i can correct an error word $\mathbf{f}^{(i)}$ if $d_H(\mathbf{f}^{(i)}) \leq d_c$, which is actually a worst case scenario that occurs only if $m_i = 1$. Although this assumption would result in a slight sacrifice in the real EPCC correction power, the marginal improvement in the bound's tightness does not justify pursuing a substantially more complicated derivation approach. To obtain the bound on the bit error probability, we need to express the error Euclidean distance enumerators as a function of the error Euclidean distance probability distribution given by (2.18). We note that the average Euclidean weight enumerator associated with the uncorrectable set of error words $\widetilde{\mathcal{C}}$ is given by:

$$\overline{T}(d_E,\widetilde{\mathcal{C}}) = \sum_{d=1}^{N} \mathbf{A}(d)\Pr(d_E|\ d,\widetilde{\mathcal{C}}) \tag{2.20}$$

While the average information input hamming distance to codewords at Euclidean distance d_E^2 is given by:

$$\overline{w}(d_E,\widetilde{\mathcal{C}}) = \frac{1}{\overline{T}(d_E,\widetilde{\mathcal{C}})} \sum_{d=1}^{N} \mathbf{A}(d)\overline{\mathbf{A}}(d)\Pr(d_E|\ d,\widetilde{\mathcal{C}}) \tag{2.21}$$

By substituting $\overline{T}(d_E,\widetilde{\mathcal{C}})$, given by (2.20), and $\overline{w}(d_E,\widetilde{\mathcal{C}})$, given by (2.21), in (2.11), we get an upper bound on the average BER of TE-EPCC as function of $\Pr(d_E|\ d,\widetilde{\mathcal{C}})$:

$$P_b \leq \sum_{d_E=d_{min}}^{\infty} \sum_{d=1}^{N} \frac{\mathbf{A}(d)\overline{\mathbf{A}}(d)\Pr(d_E|\ d,\widetilde{\mathcal{C}})}{K} Q\left(\frac{d_E}{2\sigma}\right) \tag{2.22}$$

A BER upper bound for conventional TE, of unprecoded Dicode, was found in [35] to be, ignoring the occurrence of open boundary error patterns:

$$P_b \leq \frac{1}{K} \sum_{d_E=d_{min}}^{\infty} Q\left(\frac{d_E}{2\sigma}\right) \sum_{d=1}^{N} \frac{\mathbf{A}(d)\overline{\mathbf{A}}(d)}{\begin{pmatrix} N \\ d \end{pmatrix}} \times \sum_{\substack{m=1 \\ m:\ d_E^2-2m=0 \bmod 4}}^{d} \begin{pmatrix} d-m \\ \frac{d_E^2-2m}{4} \end{pmatrix} \left(\frac{1}{2}\right)^{d-m} \begin{pmatrix} N-d \\ m \end{pmatrix} \begin{pmatrix} d-1 \\ m-1 \end{pmatrix} \quad (2.23)$$

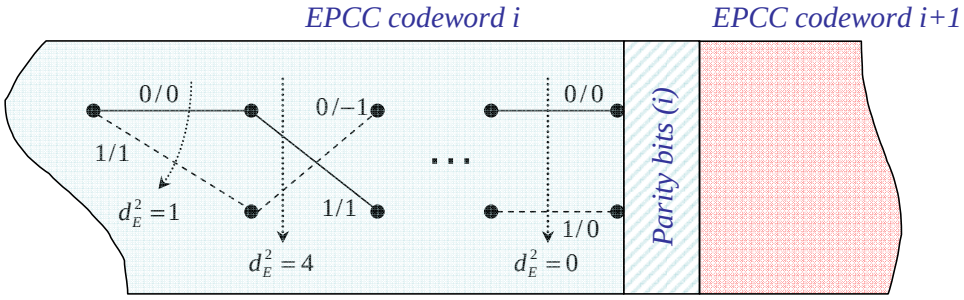


Figure 2.7: A boundary error event in the Dicode channel trellis.

Efficient Computation of Euclidean Distance Enumerator for $L > 1$ EPCC

A more compact and efficient method is derived here to evaluate the multiple summations in equations (2.19) and (2.17), which are used to compute the BER bound in (2.22). But first, we will reincorporate the effect of boundary error patterns in the Euclidean weight enumerator, a task made easier by this more efficient method. Figure 2.7 shows an error pattern at the boundary of an EPCC subcode's data and parity fields. The boundary is chosen this way since we do not count error patterns in the EPCC codeword's parity, where this field is discarded before passing the decoded data to the outer interleaved RSCC decoder. By examining the trellis we note that boundary errors contribute a Euclidean distance of $d_E^2 = 1 + 4 \times b_{cr}$ in general compared to $d_E^2 = 2 + 4 \times b_{cr}$ for closed error patterns that are totally encapsulated by the current subcode i. Further more, there are now

only $\begin{pmatrix} N_c - d_i \\ m_i - 1 \end{pmatrix}$ ways by which the disjoint m_i error patterns of error word $\mathbf{f}^{(i)}$ can be arranged in the current subcode i, given the subcode has a boundary error. The reduced number of possible arrangements of m_i is due to the given fact that the last error pattern occurs only at the boundary for subcodes assumed to have a boundary error. Including error words $\mathbf{f}^{(i)}$ that span the boundary, the total number of possible error pattern arrangements becomes $\begin{pmatrix} N_c - d_i + 1 \\ m_i \end{pmatrix}$.

Next, We define a probability enumerator for subcode i for all possible values of the parameters d_i, m_i and μ_i, that is given by the multinomial

$$\begin{aligned} \mathbf{\Lambda}(\mathrm{D}, \mathrm{M}, \Upsilon;\ m_{max}, d_{max}) &= 1 + \sum_{\mu_i=0}^{1} \sum_{d_i=1}^{d_{max}} \sum_{m_i=1}^{\min(d_i, m_{max})} \\ &\left(\frac{1}{2}\right)^{d_i - m_i} \begin{pmatrix} N_c - d_i \\ m_i - \mu_i \end{pmatrix} \begin{pmatrix} d_i - 1 \\ m_i - 1 \end{pmatrix} \mathrm{D}^{d_i} \mathrm{M}^{m_i} \Upsilon^{\mu_i} \end{aligned} \tag{2.24}$$

where the $\mathrm{D}^0\mathrm{M}^0\Upsilon^0 = 1$ monomial term corresponds to the case when there are no errors in the specified subcode, and $\mu_i = \{0, 1\}$ is the number of boundary errors per subcode. As a result, the probability enumerator for the entire interleave composed of L EPCC subcodes is given by

$$\mathbf{\Lambda}^L(\mathrm{D}, \mathrm{M}, \Upsilon;\ m_{max}, d_{max})$$

given that only d_{max}-weight error words $\mathbf{f}^{(i)}$ composed of m_{max} disjoint error patterns can occur per EPCC subcode, where d_{max} and m_{max} are unbounded from above if EPCC correction is turned off. The advantage of this approach is that polynomial multiplication, or the more general multinomial multiplication, can be performed efficiently by symbolic manipulators, such as Maple™. Utilizing the compact, and efficient to compute, probability

enumerator, we can now express the bound on the bit error rate of TE-EPCC as:

$$P_b \leq \frac{1}{K} \sum_{d_E=d_{min}}^{\infty} Q\left(\frac{d_E}{2\sigma}\right) \sum_{d=1}^{N} \frac{\mathbf{A}(d)\overline{\mathbf{A}}(d)}{\begin{pmatrix} N \\ d \end{pmatrix}} \sum_{\mu=0}^{L} \sum_{\substack{m=1 \\ m:\ d_E^2-2m+\mu=0 \bmod 4}}^{d} \begin{pmatrix} d-m \\ \frac{d_E^2-2m+\mu}{4} \end{pmatrix} [\mathbf{\Lambda}^L(\mathrm{D}, \mathrm{M}, \Upsilon;\ \infty, \infty)]_{d,m,\mu} - \sum_{\substack{m=1 \\ m:\ d_E^2=2m-\mu}}^{d} [\mathbf{\Lambda}^L(\mathrm{D}, \mathrm{M}, \Upsilon;\ m_c, d_c)]_{d,m,\mu} \tag{2.25}$$

where the probability enumerator for a correctable EPCC codeword is approximated by

$$\mathbf{\Lambda}^L(\mathrm{D}, \mathrm{M}, \Upsilon;\ m_c, d_c),$$

for an EPCC of maximum correction power m_c per subcode.

2.4.3 Interleaver Gain Exponent of TE-EPCC

To gain insight into how EPCC enhances TE performance, we pursue an analytic approach to study the mechanism by which EPCC reduces the multiplicity of low Euclidean distance errors. By further examining (2.11), it is deducible that there are two approaches to lower the BER. In the first approach, one can increase the minimum Euclidean distance between error words, an approach pioneered by Ungerboeck in the seminal work [55], see also [56, 57], and termed trellis-coded modulation. In the second approach, rather than increasing the argument of the error function, one can substantially reduce the multiplicity of low Euclidean distance errors, as in the more recent turbo coding approach pioneered by Berrou *et al.* in [21]. In turbo coding, the coefficients of the error function for low Euclidean distances are an inverse function of the interleaver size, N. For this reason, turbo coding gain is often referred to as interleaver gain. In more detail, for the frequency of low weight errors to asymptotically approach zero as the interleave size tends to infinity, the exponent

of the interleaver length should be less than zero. Therefore, we can argue for the advantage of incorporating EPCC in TE, by showing how it works to decrease the exponent of N^{α} well below zero, especially for low Euclidean distance errors. We call α the interleaver gain exponent. First, we isolate the exponent of N in the expression of BER for TE and TE-EPCC. The BER expression of conventional TE, when accounting for the occurrence of boundary errors is:

$$P_b \leq \frac{1}{K}\sum_{d_E=1}^{\infty} Q\left(\frac{d_E}{2\sigma}\right)\sum_{d=2}^{d_T}\frac{\mathbf{A}(d)\overline{\mathbf{A}}(d)}{\binom{N}{d}} \sum_{\mu=0}^{1}\sum_{\substack{m=1 \\ m:\ d_E^2-2m+\mu=0 \bmod 4}}^{d}\binom{d-m}{\frac{d_E^2-2m+\mu}{4}}\left(\frac{1}{2}\right)^{d-m}\binom{N-d}{m-\mu}\binom{d-1}{m-1} \tag{2.26}$$

where $d_T \ll N$ is the truncated maximum error weight. We truncated the Hamming error weight d since large weight errors correspond to larger Euclidean distances which have extremely marginal contribution to the BER. To produce an expression for the upper bound on BER with isolated powers of N, and at the same time preserve it as an upper bound, we replace the binomial in the denominator by the lower bound, [34]:

$$\binom{N}{d} > \frac{(N-d+1)^d}{d!} \simeq \frac{N^d}{d!}.$$

Moreover, to replace the binomial in the numerator with an upper bound that is also a power of $N-d+1$, we first express it as:

$$\binom{N-d}{m-\mu} = \frac{m-\mu+1}{N-d+1}\binom{N-d+1}{m-\mu+1}$$

and then, we employ the upper bound, [34]:

$$\binom{N-d+1}{m-\mu+1} < \frac{(N-d+1)^{m-\mu+1}}{(m-\mu+1)!} \simeq \frac{N^{m-\mu+1}}{(m-\mu+1)!}$$

These bounds are tight when N is large, and $d, m \ll N$, which holds true in our case. Also we can upper bound the Q function by:

$$Q\left(\frac{d_E}{2\sigma}\right) \leq \frac{1}{2} e^{-\frac{d_E^2}{4\sigma^2}}.$$

Substituting these approximate bounds in the BER upper bound in (2.26), we get the looser but useful bound:

$$P_b \;<\; \frac{1}{2K} \sum_{d_E=1}^{\infty} \sum_{d=2}^{d_T} \sum_{\mu=0}^{1} \sum_{\substack{m=1 \\ m:\ d_E^2-2m+\mu=0 \bmod 4}}^{d} \mathbf{B}_{d_E,d,m,\mu} N^{m-\mu-d} e^{-\frac{d_E^2}{4\sigma^2}} \tag{2.27}$$

where $\mathbf{B}_{d_E,d,m,\mu}$ is given by:

$$\mathbf{B}_{d_E,d,m,\mu} = \mathbf{A}(d)\overline{\mathbf{A}}(d) \frac{d!}{(m-\mu)!} \left(\frac{1}{2}\right)^{d-m} \begin{pmatrix} d-m \\ \frac{d_E^2-2m+\mu}{4} \end{pmatrix} \begin{pmatrix} d-1 \\ m-1 \end{pmatrix} \tag{2.28}$$

For the sake of mathematical tractability, and without any loss in generality, we study the interleaver gain exponent of $L = 1$ TE-EPCC, i.e. single EPCC subcode per interleave. Utilizing the same approximations above in the BER bound of TE-EPCC for $L = 1$ we get the expression:

$$\begin{aligned} P_b \;<\; & \frac{1}{2K} \sum_{d_E=1}^{\infty} e^{-\frac{d_E^2}{4\sigma^2}} \sum_{\mu=0}^{1} \\ & \left[\sum_{d=2}^{d_T} \sum_{\substack{m=1 \\ m:\ d_E^2-2m+\mu=0 \bmod 4}}^{d} \mathbf{B}_{d_E,d,m,\mu} N^{m-\mu-d} \right. \\ & \left. - \sum_{d=2}^{\min(d_T,d_c)} \sum_{\substack{m=1 \\ m:\ d_E^2=2m-\mu}}^{\min(d,m_c)} \mathbf{B}_{d_E,d,m,\mu} N^{m-\mu-d} \right] \end{aligned} \tag{2.29}$$

The expression in (2.29) is just the expression in (2.27) with those terms that are correctable by EPCC subtracted. By identifying the maximum exponent of the interleaver length N in (2.29) and (2.27), we can compare the asymptotic BER of TE and TE-EPCC in the limit

of large interleaver size. We list the maximum interleaver gain exponent per Euclidean error distance, d_E^2, for TE and TE-EPCC($d_c = 10$, $m_c = 3$, $L = 1$) in Table 2.4 for $d_E^2 = \{1, \ldots, 4\}$, in Table 2.5 for $d_E^2 = \{5, 6\}$, and in Table 2.6 for $d_E^2 = 7$. We also list for each d_E^2, the generating error patterns and their corresponding parameters d, m, and μ. In addition, under each interleaver gain exponent, we list in between brackets the corresponding multiplicative coefficient $\mathbf{B}_{d_E,d,m,\mu}$, excluding the term $\mathbf{A}(d)\overline{\mathbf{A}}(d)$ relating to the outer RSCC Hamming error weight distribution.

First, we note that for unprecoded TE, the interleaver gain exponents are all negative for $d_E^2 = 1$ to $d_E^2 = 3$, which are the terms that dominate the BER for medium to high SNR. Second, we note that the error patterns, for this same range of error Euclidean distances, up to $d = 10$ all belong to the dominant error class. As a result, TE-EPCC manages to substantially decrease the interleaver gain exponent by a factor of N^{-9}. Also, for $d_E^2 = 4$, where TE does not achieve any interleaver gain, TE-EPCC has an impressive interleaver gain exponent of N^{-9}.

The extremely low exponents suggest that TE-EPCC will have large gain even for relatively short interleavers, and would thus appear insensitive to the interleaver size. At the same time, for such short interleavers, TE would considerably suffer in terms of turbo gain. These conclusions will be numerically demonstrated in the next section by evaluating the BER bound for interleavers as short as 100 bits. Furthermore, although $\mathbf{B}_{d_E,d,m,\mu}$ is significantly larger in TE-EPCC compared to TE for the same d_E^2, the term $\mathbf{B}_{d_E,d,m,\mu}N^{m-\mu-d}$ is still several orders of magnitude lower for TE-EPCC compared to TE.

Although less important, we also show the interleaver gain for higher error Euclidean distances in Table 2.5 and Table 2.6. Most notably, TE-EPCC($d_c = 10$, $m_c = 3$, $L = 1$) corrects errors belonging to the dominant error class for $d_E^2 = 5$ and $d_E^2 = 6$, lowering, in the process, the maximum interleaver gain exponent by a factor of N^{-1}, a turbo gain that becomes more substantial for large interleavers. Actually, for $d_E^2 = 6$, TE possess no interleave gain, while TE-EPCC BER is dominated by the non-targeted set of non-dominant

Table 2.4: Interleaver gain exponent of conventional unprecoded TE vs TE-EPCC, $d_E^2 = \{1, 2, 3, 4\}$.

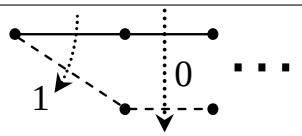
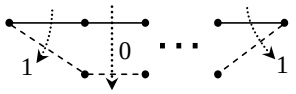
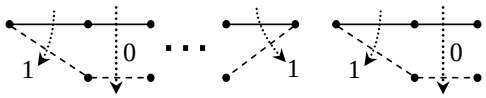
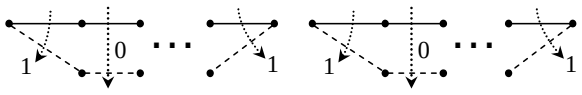
$d_E^2 = 1$	Error pattern classes	TE	TE-EPCC
$m = 1$ $\mu = 1$ $d = 2 \rightarrow d_T$		N^{-2} (1)	N^{-11} $\left(\frac{155925}{4}\right)$
$d_E^2 = 2$	Error pattern classes	TE	TE-EPCC
$m = 1$ $\mu = 0$ $d = 2 \rightarrow d_T$		N^{-1} (1)	N^{-10} $\left(\frac{155925}{4}\right)$
$d_E^2 = 3$	Error pattern classes	TE	TE-EPCC
$m = 2$ $\mu = 1$ $d = 2 \rightarrow d_T$		N^{-1} (2)	N^{-10} (779625)
$d_E^2 = 4$	Error pattern classes	TE	TE-EPCC
$m = 2$ $\mu = 0$ $d = 2 \rightarrow d_T$		N^{0} (1)	N^{-9} $\left(\frac{779625}{2}\right)$

Table 2.5: Interleaver gain exponent of conventional unprecoded TE vs TE-EPCC, $d_E^2 = \{5, 6\}$.

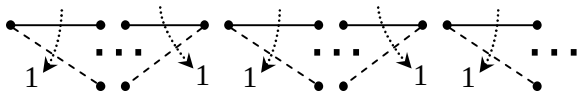
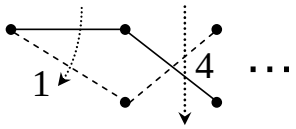
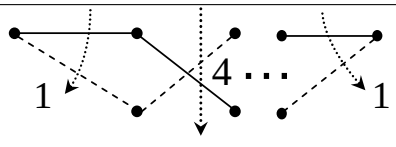
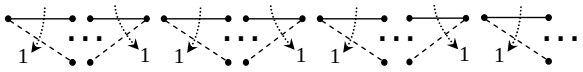
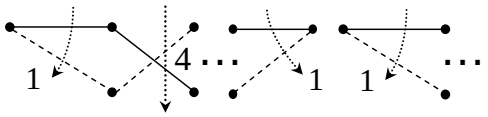
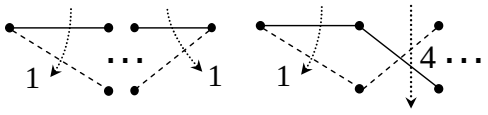
$d_E^2 = 5$	Error pattern classes	TE	TE-EPCC
$m = 3$ $\mu = 1$ $d = 3 \rightarrow d_T$		N^{-1} (3)	N^{-9} (3508313)
$d_E^2 = 5$	Error pattern classes	TE	TE-EPCC
$m = 1$ $\mu = 1$ $d = 2 \rightarrow d_T$		N^{-2} (1)	N^{-2} (1)
$d_E^2 = 6$	Error pattern classes	TE	TE-EPCC
$m = 3$ $\mu = 0$ $d = 3 \rightarrow d_T$		N^{0} (1)	N^{-8} (1169438)
$d_E^2 = 6$	Error pattern classes	TE	TE-EPCC
$m = 1$ $\mu = 0$ $d = 2 \rightarrow d_T$		N^{-1} (1)	N^{-1} (1)

Table 2.6: Interleaver gain exponent of conventional unprecoded TE vs TE-EPCC, $d_E^2 = 7$.

$d_E^2 = 7$	Error pattern classes	TE	TE-EPCC
$m = 4$ $\mu = 1$ $d = 4 \rightarrow d_T$		N^{-1} (4)	$[N^{-1}]_{m_c=3}$ (4) $[N^{-8}]_{m_c=4}$ (6237000)
$d_E^2 = 7$	Error pattern classes	TE	TE-EPCC
$m = 2$ $\mu = 1$ $d = 3 \rightarrow d_T$		N^{-2} (6)	N^{-2} (6)
$d_E^2 = 7$	Error pattern classes	TE	TE-EPCC
$m = 2$ $\mu = 1$ $d = 3 \rightarrow d_T$		N^{-2} (6)	N^{-2} (6)

errors that result in the exponent N^{-1}, achieving interleaver gain thus still. On the other hand, TE-EPCC($d_c = 10$, $m_c = 3$, $L = 1$) would offer no advantage when $d_E^2 = 7$, since the number of targeted dominant error patterns, m, is larger than the maximum multiple-error-pattern correction capability of $m_c = 3$. However, TE-EPCC($d_c = 10$, $m_c = 4$, $L = 1$) manages to reduce the maximum interleaver gain exponent by a factor N^{-1}. Finally, all conclusions based on the interleaver gain exponent will be substantiated by a numerical study of the Euclidean distance error enumerator and the BER bound in the next section.

2.5 Numerical Analytic and Simulation Results

Utilizing the analytic approximation of the BER of conventional TE and TE-EPCC systems, we study the relative performance of these systems in a variety of channel conditions with a rate penalty proportional to $10\log_{10}\frac{1}{R}$, where R is the code rate. The log of the distribution of error Euclidean distance is shown in Figure 2.9 for conventional TE and TE-EPCC, Figure 2.9 also includes the error distribution for precoded TE. $\log\overline{T}(d_E)$ is calculated for a TE with $K = 4096$, rate $1/2$ base constituent RSCC, punctured to rate $R = 8/9$ with generator polynomial connections $(31, 33)$ in octal format, and $L = 7$ EPCC with $m_c = 3$ and $d_c = 10$. Each EPCC subcode is a $(630, 616)$ systematic cyclic code of rate 0.98 shortened to achieve $L = 7$. The results for the precoded case were generated by the expression of $\overline{T}(d_E)$ in [35] for a Dicode channel and a $\frac{1}{1\oplus D}$ precoder. We omit the derivation of this expression here, but comment on the operation of the precoder and show its performance compared to unprecoded systems. Through the study of the error Euclidean distance distribution we note that the dominant contributor to the estimated BER is determined by the Euclidean distance and corresponding multiplicity of codewords at that minimum distance that together yield the largest spectral component in the union bound. Hence, examining the trellis of the unprecoded Dicode channel in Figure 2.4, and the trellis of its precoded counterpart in Figure 2.8, we conclude that the reason why unprecoded channels have high BER in the waterfall region, as will be shown by the union bound and simulations, is that the unprecoded trellis has paths that correspond to different code bits but are at 0 Euclidean distance, meaning that, long error events have a high probability of generating low Euclidean distances. On the other hand, this behavior is eliminated in the precoded Dicode trellis, where paths associated with different coded bits do accumulate Euclidean distance. Thus, for the precoded trellis, diverging paths corresponding to errors with large Hamming weight result in accumulation of Euclidean distance and reduction of the contribution of these high weight errors to the BER union bound. At the same time, the average number of error words that generate low Euclidean distance errors is proportionally

reduced. In other words, only low Hamming weight errors generate low Euclidean distance discrepancies from the genie path. This means the precoded Dicode channel would exhibit larger interleaver gain compared to the unprecoded channel. However, for higher SNR, at the error floor region, the contribution of Euclidean distance 2 becomes stronger, and as seen in Figure 2.9, the average number of Hamming weight 2 errors that generate $d_E^2 = 2$ is more for precoded compared to unprecoded Dicode. On the other hand, EPCC concentrates on low Euclidean distances, reducing their frequency substantially up to $d_E^2 = 6$. This results in improved BER performance in both of the waterfall and error floor regions compared to conventional TE.

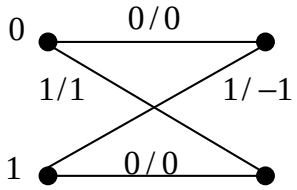


Figure 2.8: Trellis section for the precoded Dicode channel.

In Figure 2.10, we simulate BER of conventional TE with the precoded and unprecoded Dicode channel and TE-EPCC for unprecoded Dicode, and compare it with the estimated BER bound computed for the same parameters as in Figure 2.9. TE-EPCC is decoded via the practical soft decoder described earlier in the chapter, where we implement 5 turbo iterations of unprecoded TE and 10 turbo iterations of precoded TE and TE-EPCC, using up to 90 test patterns in the list decoder of TE-EPCC. The number of turbo iterations is chosen for each system based on the point beyond which the turbo gain saturates. The figure shows that TE-EPCC can achieve up to 2 dB SNR gain with respect to unprecoded TE and more than 1 dB with respect to precoded TE at a BER of 10^{-6}. While the estimated bounds predict 1 dB SNR gain with respect to unprecoded TE, and 0.3 dB worse performance with respect to precoded TE. The discrepancy between the BER bound and simulation results arises from imperfect uniform interleaving in the practical decoders. Nonetheless, the gain gap between TE-EPCC and conventional TE is always worse than predicted by the bound,

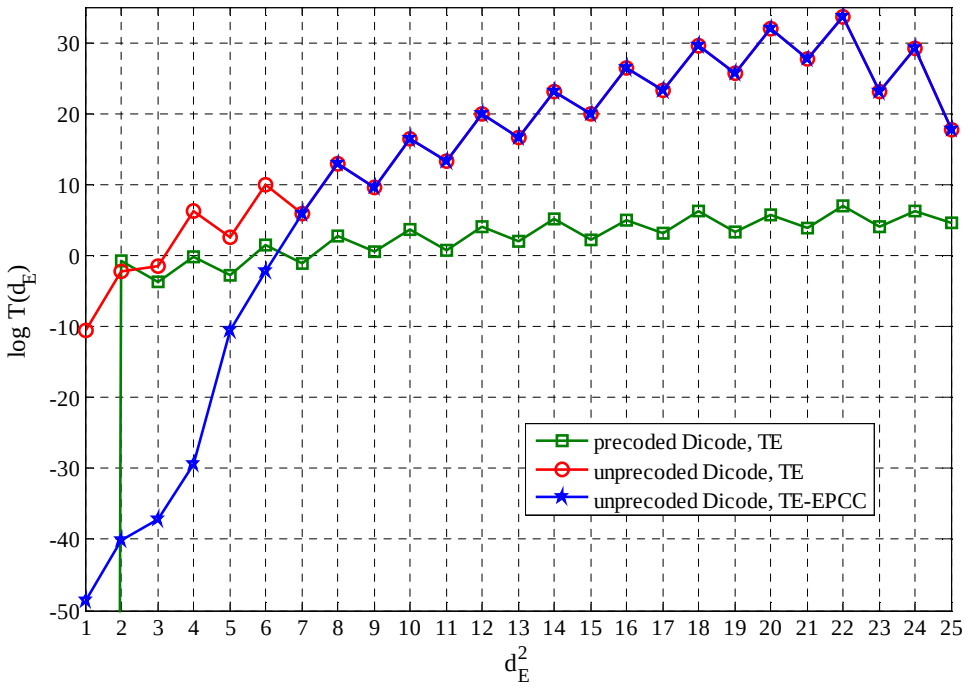


Figure 2.9: $\log \overline{T}(d_E)$ for various TE systems, and $(31, 33)$ RSCC.

where this is mainly attributed to the higher sensitivity of conventional TE to the interleaver design compared to TE-EPCC, an argument based on the interleaver gain exponent of both systems. We also note from the figure that at a lower BER of 10^{-8}, an impressive gain of 3 dB is furnished by TE-EPCC.

The effect of interleaver size is studied next. In Figure 2.11 we compare the BER bound of conventional TE with the precoded and unprecoded Dicode channel and TE-EPCC for unprecoded Dicode, for interleaver lengths $N = \{100, 200, 400, 600, 800, 1000, 2000\}$, punctured rate $R = 8/9$ RSCC with connections $(7, 5)$ in octal format, and EPCC with $m_c = 3$ and $d_c = 10$, where the arrows indicate the direction of increasing N. To obtain shorter EPCC code lengths, serially concatenating one EPCC subcode per RSCC interleave, the EPCC code length is shortened from the long $(630, 616)$ EPCC at the same level of redundancy. While to support interleaver sizes above 630, we duplicate EPCC subcodes and use shortening to fit fractions of EPCC subcodes in one interleave. For instance, we

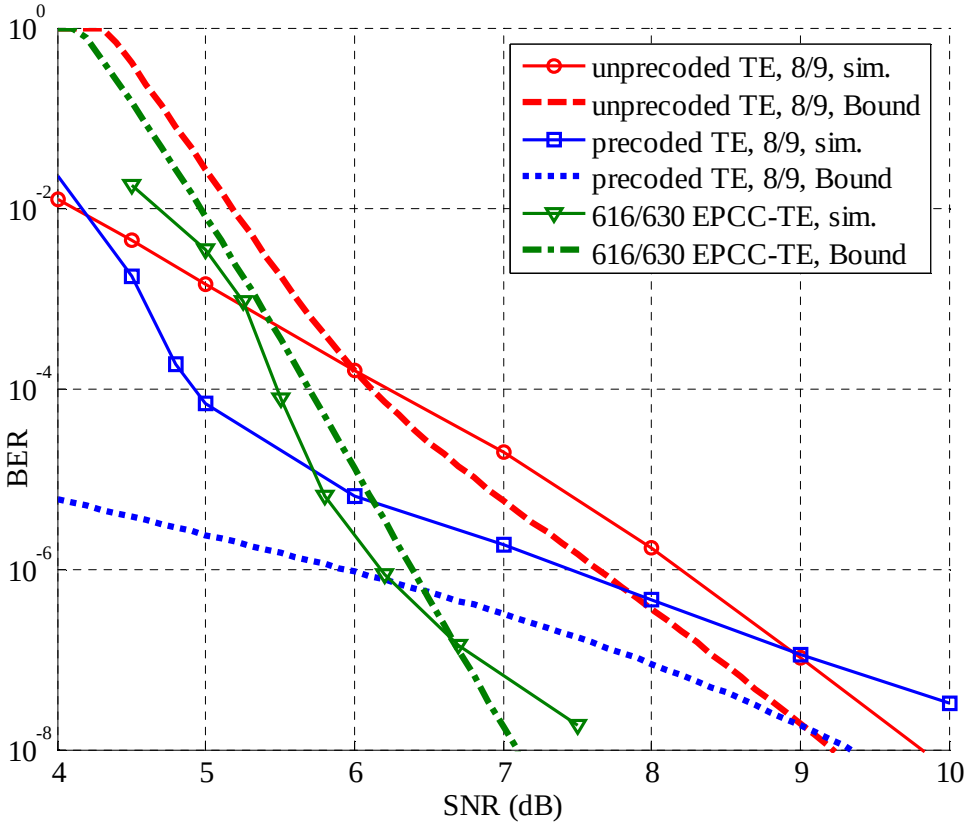


Figure 2.10: Simulated vs bounded BER for various TE systems, and $(31, 33)$ RSCC.

implement $(100, 86)$ EPCC of rate 0.86 for interleaver length $N = 100$, and long $(630, 616)$ EPCC + shortened $(398, 384)$ EPCC for $N = 1000$. As the interleaver size decreases, turbo gain decreases accordingly. Also, since we maintain the same number of parity bits as the codeword length shrinks, more rate penalty is incurred. On the other hand, the multiplicity of error patterns reflected by $m > 1$ reduces. These conflicting effects even out, and the performance of TE-EPCC remains almost the same even for shorter interleavers. Furthermore, since the turbo gain of conventional TE diminishes for shorter interleavers, the gain furnished by TE-EPCC grows from about 1 dB at BER 10^{-6} and $N = 2000$ bits to an impressive 2.5 dB for $N = 100$ bits. Therefor, for delay-sensitive applications, TE-EPCC has an edge on more conventional TE systems. In a similar study of the performance of $(7, 5)$ RSCC based TE versus $(5, 7)$ RSCC based TE, it was shown in [31] that the perfor-

mance of TE using $(5,7)$ RSCC is considerably worse than $(7,5)$ due to the larger number of Hamming weight 2 error sequences resulting in higher frequency of $d_E^2 = 2$ errors. This can be demonstrated quantitatively by the evaluation of $\overline{T}(d_E)$ of both systems, which is shown on the same graph for all TE designs in Figure 2.13, where we compare $\log \overline{T}(d_E)$ for interleaver size $N = 600$, punctured rate $R = 8/9$ RSCC with connections $(7,5)$ and $(5,7)$, and $L = 1$ EPCC with $m_c = 3$ and $d_c = 10$. In this figure, we observe that the frequency of low Euclidean distances is much higher for $(5,7)$ RSCC based systems compared to TE using $(7,5)$ RSCC. Nonetheless, TE-EPCC is equally successful in reducing the effect of low Euclidean distance errors for both systems, delivering 2 dB SNR gain at 10^{-6} BER compared to $(5,7)$ RSCC based TE for interleaver size $N = 100$, as shown in Figure 2.12 for the same set of parameters used in Figure 2.11.

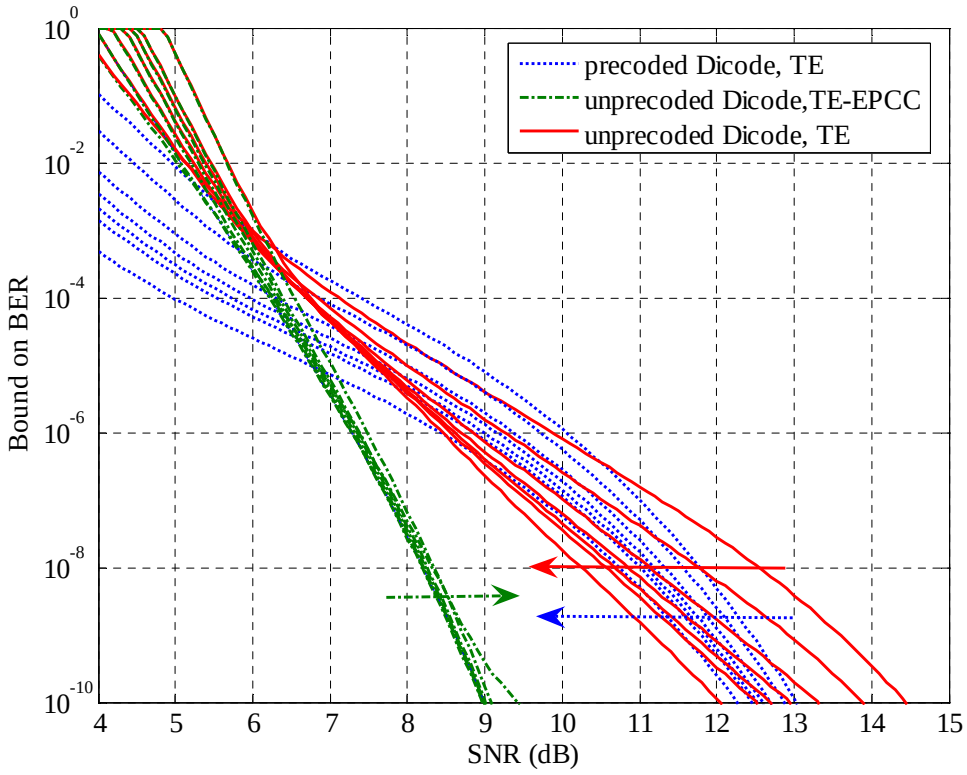


Figure 2.11: BER of various TE systems for different interleaver lengths and $(7,5)$ RSCC.

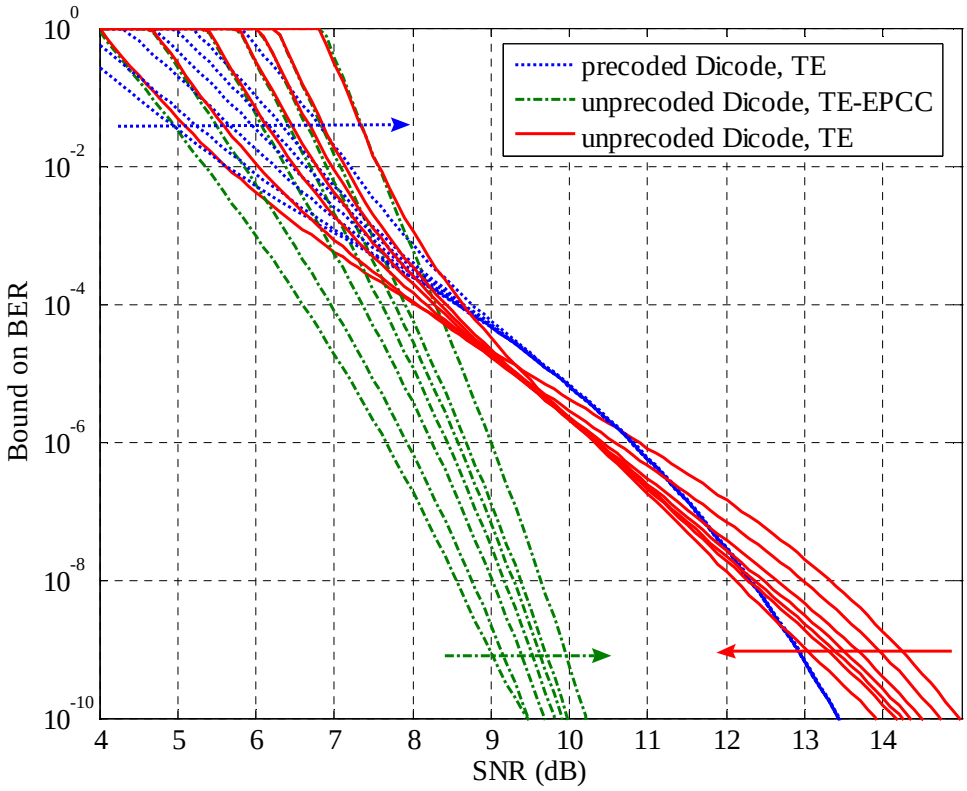


Figure 2.12: BER of various TE systems for different interleaver lengths and $(5, 7)$ RSCC.

The performance of TE-EPCC can be further improved by increasing its multiple error correction capability m_c, per subcode. However, the complexity of the decoder would increase accordingly as more test words have to be constructed in the list decoder. Figure 2.14, compares the BER of TE with the precoded and unprecoded Dicode channel and TE-EPCC for unprecoded Dicode, for interleaver length $N = 1200$, punctured $R = 8/9$ RSCC with connections $(7, 5)$, and $L = 1$ EPCC with maximum correction capability increased from $m_c = 1$ to $m_c = 10$ and $d_c = 10$. The arrow indicates the direction of increasing m, where the SNR gain of TE-EPCC at 10^{-6} grows from a few tenth of dB for maximum correction capability $m_c = 1$, to more than 2 dB for $m_c = 10$ TE-EPCC. Another design method to increase the correction capability of TE-EPCC, without considerable increase in complexity, is to use $L > 1$ EPCC subcodes per interleave. Figure 2.15

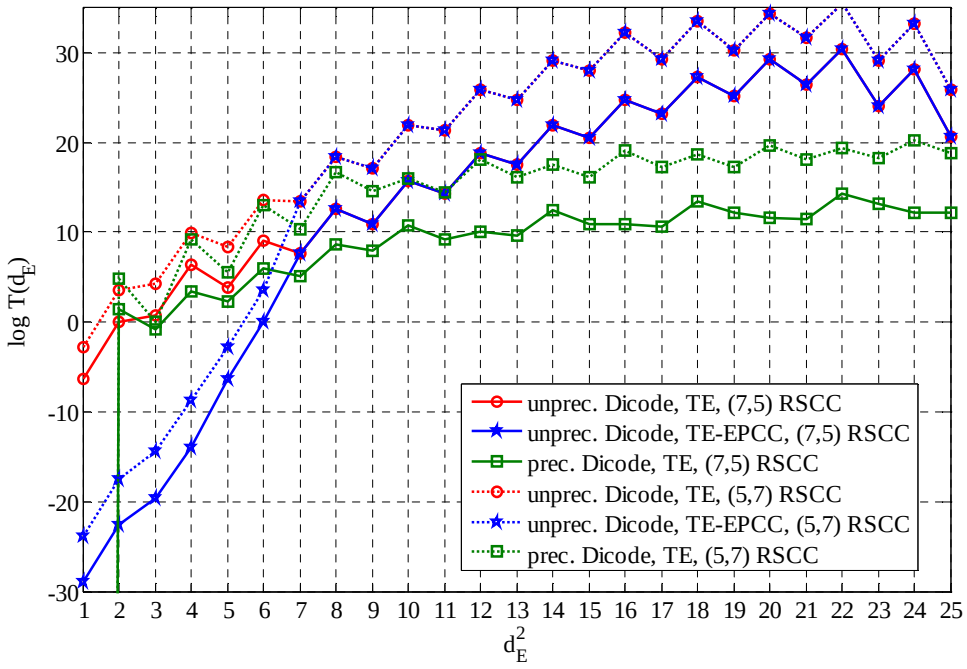


Figure 2.13: $\log \overline{T}(d_E)$ for various TE systems, and $(7, 5)$ RSCC vs $(5, 7)$ RSCC.

shows the BER for TE-EPCC for interleaver length $N = 1200$, punctured-rate $8/9$ RSCC with connections $(7, 5)$, and EPCC with different combinations of m_c and L, and $d_c = 10$. We again obtain shorter length EPCC codes by employing the concept of code shortening. As the number of subcodes increases per interleave, the correction capability improves at an increasingly slower pace. This is due to higher level of redundancy required for shorter EPCC to maintain the maximum correction capability m_c of longer EPCC. For instance, to maintain the correction capability at a code length of 30 bits, i.e. $L = 40$ and $N = 1200$, a shortened EPCC of rate 0.53 would incur a staggering rate penalty of 2.7 dB. A concatenation approach that avoids the rate penalty of shorter EPCC will be discussed later in chapter 4.

We also wish to study TE-EPCC advantage at various total system rates and distributions of redundancy between the outer RSCC and inner EPCC subcode. In Figure 2.16, we compare the simulated BER of conventional unprecoded TE and TE-EPCC for unprecoded

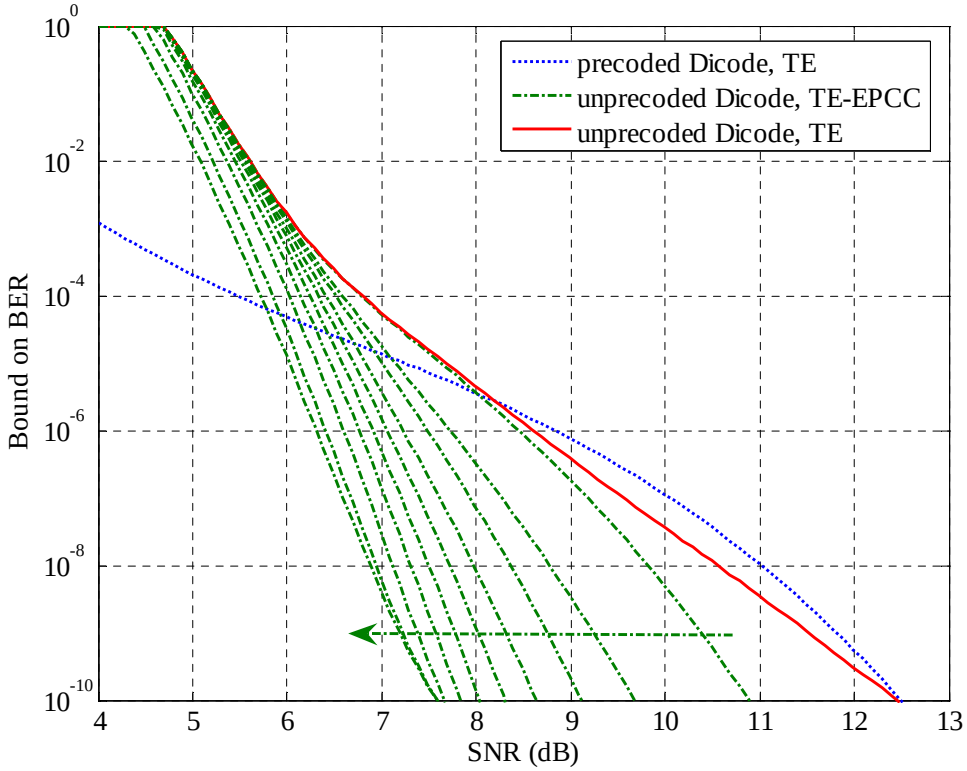


Figure 2.14: Comparison of TE-EPCC systems with different m_c.

Dicode, for interleaver length $N = 4312$, different punctured-rate RSCC with connections $(7, 5)$, and EPCC with $m_c = 3$ and $d_c = 10$. The results show that TE-EPCC composed of either $L = 7$ $(630, 616)$ EPCC or $L = 22$ $(210, 199)$ EPCC concatenated to rate $\frac{5}{6}$ TE, achieve the same BER in the error floor region. Furthermore, they both outperform comparable rate conventional TE, with $L = 22$ TE-EPCC furnishing a gain of 1.5 dB with respect to rate $\frac{3}{4}$ TE, and $L = 7$ TE-EPCC delivering similar gain over rate $\frac{5}{6}$ TE. For a complete investigation of a wide range of coding rates, we plot the minimum SNR required to achieve a BER of 10^{-7} for punctured coding rates from $\frac{2}{3}$ to $\frac{9}{10}$, comparing conventional unprecoded and precoded TE to TE-EPCC. Such a comparison is shown in Figure 2.17 for interleaver length $N = 1200$, different punctured-rate RSCC with connections $(7, 5)$, and $L = 1$ EPCC with $m_c = 3$ and $d_c = 10$. We conclude from the results that TE-EPCC delivers a uniform gain of 1.5 dB for puncturing rates above $\frac{3}{4}$. The abnormal peak in BER

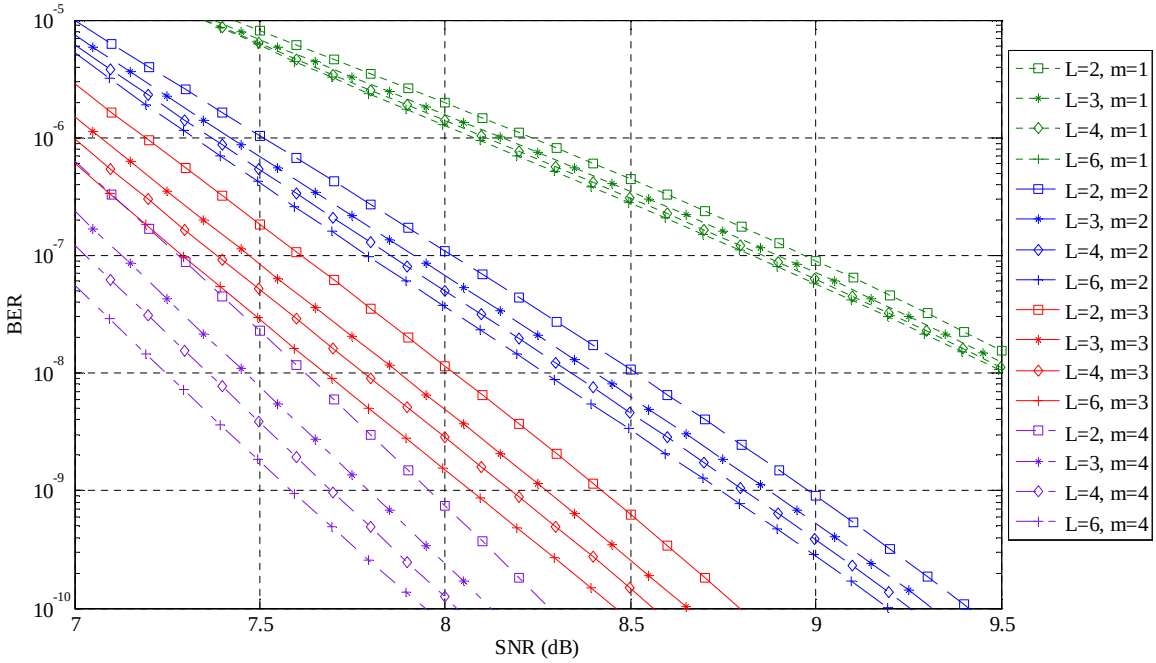


Figure 2.15: BER of TE-EPCC for different m_c and L combinations.

for puncturing rate $\frac{6}{7}$ is due to the particular choice of puncturing table. Even so, TE-EPCC seems to be more resilient to the choice of puncturing table, as would be concluded from the observation that the turbo gain gap rises to 2 dB at these troublesome punctured rates. The reason why precoded TE outperforms TE-EPCC for puncturing rates $\frac{2}{3}$ and $\frac{3}{4}$ can be explained by examining Table /refch2:Table:Ad, where we can see that the outer RSCC does not generate Hamming weight 2 errors for those puncturing rates. Thus, since the BER performance of precoded TE is dominated by such errors in the floor region, its BER is significantly improved surpassing TE-EPCC at those rates.

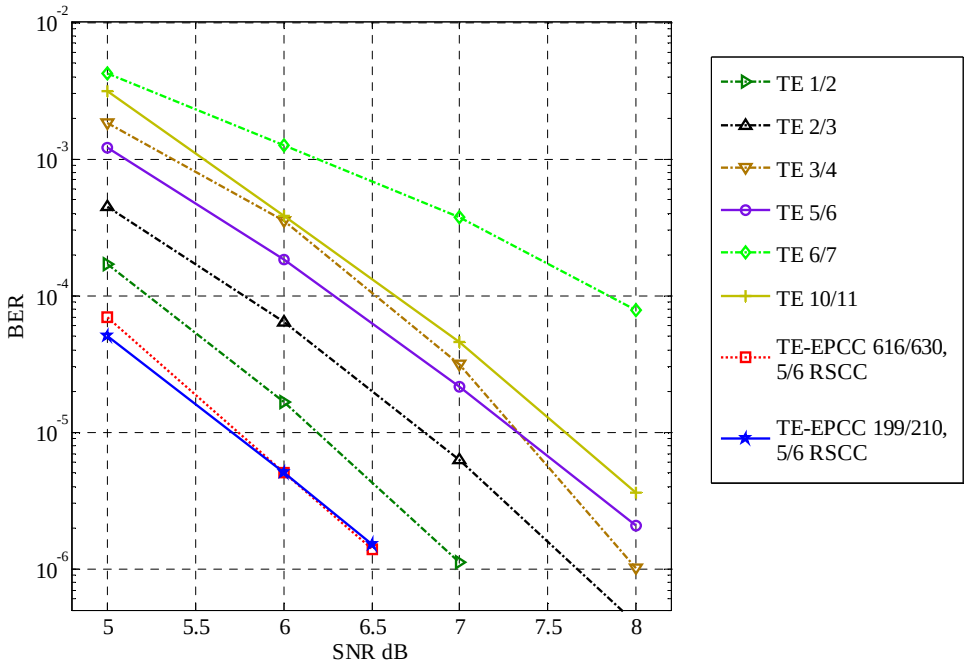


Figure 2.16: Simulation of TE-EPCC and conventional TE for various rates.

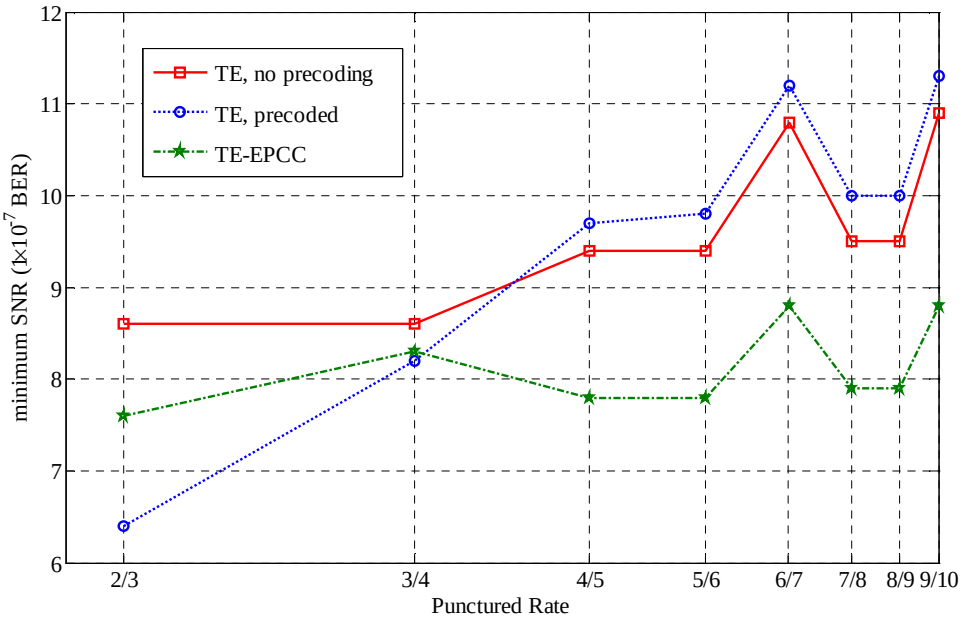


Figure 2.17: Minimum SNR required to achieve 10^{-7} BER for various TE systems.

Chapter 3

EPCC-based Iterative Equalization of Magnetic Recording Channels

We consider the application of the error-pattern correcting code in turbo-equalization of magnetic recording channels, where the channel-matched EPCC decoder exchanges soft information iteratively with the convolutional code decoder. To approach maximum likelihood (ML) decoding of the proposed turbo systems, we utilize soft iterative decoding of the component codes, in which, EPCC is decoded via the soft-in soft-out (SISO) list decoder of the algebraic single error-pattern decoder discussed in the previous chapter. The EPCC SISO list decoder receives extrinsic information coming from the channel detector as a priori information in addition to channel observations in calculating error pattern a posteriori probabilities, where this process is accomplished by a bank of error-pattern-matched local correlators. The most probable of these error patterns are used to perform list decoding through the following process:

1. The test error word list is generated by inserting the most probable combination of local error patterns to the channel detector output.

2. An array of parallel single-pattern correcting decoders decodes the test words to produce a list of valid codewords.

3. The list of candidate codewords is used to generate bit-level decisions along with their reliabilities to be passed as a priori information to the SISO decoders of other component codes.

In the previous chapter, we have shown that reallocating a small portion of the outer convolutional code redundancy to the inner high rate EPCC results in an improved system performance with manageable added complexity. In this chapter, This performance advantage is investigated in high density magnetic recording channels corrupted by a noise mixture that is dominated by data-dependent medium noise. For such channels, the bank of error-pattern-matched correlators include pattern-dependent noise prediction (PDNP) -not requiring trellis expansion- which is run on the ML word released by the noise predictive channel detector.

We evaluate the BER performance of TE-EPCC, and comment on the potential performance gain after concatenation with an outermost Reed-Solomon (RS) code in hopes of removing the TE error floor. A performance comparison with conventional TE as well as an LDPC-coded system is also presented in high density magnetic recording channels.

3.1 The Magnetic Recording Channel

The magnetic recording write-read process is modelled by an inter-symbol-interference bandlimited communications channel that is plagued by various channel noises and impairments. In the write process of the current recording paradigm, bits are stored on the media through saturated-level recording. This means that the magnetization levels in the recording media can exist in only two states, which translates to binary-constrained input communications in the corresponding channel model. The binary-input constraint precludes fancy transmitter-side methods to approach the bandlimited channel capacity, including SNR water-filling, optimal channel precoding, and continuous trellis-coded modulation. Hence, recording read-channel designers resort to powerful error correction coding tech-

niques to approach the i.u.d. binary-input constrained channel capacity. Furthermore, in the read process, binary bits correspond to magnetization level transitions. Therefore, the recording channel is often referred to as being "transition-response fixed", i.e. the width of the transition response is fixed irrespective of the symbol period. Hence, to store the same number of user bits at a lower code rate, bit transitions have to be placed more closely on the recording media. The result is increased inter-symbol interference that can be modelled by an SNR degradation that is inversely proportional to the coding rate as in $10 \log_{10} \frac{1}{R^2}$, which is referred to as the rate penalty of the code. This severe SNR degradation, a 6 dB SNR loss at $R = \frac{1}{2}$, implies that the ECC designer is limited to the design space of very high code rates. This high coding rate constraint complicates system design further due to the stringent standard on storage fidelity. Specifically, the fact that user-sector rereads are expensive in terms of delay and effectiveness, since media impairments persist between rereads, means that the ECC is expected to deliver sector error rates as low as $10^{-13} \sim 10^{-15}$ in normal operation. The search for high rate ECC that can guarantee such low SER as close as possible to the channel capacity is still an active area of research. Hopefully, the work in this thesis serves as a contribution in that direction. In addition, the ECC decoder has to handle various bursty channel impairments characteristic of the recording system, including media defects and thermal asperities. For such burst impairment, symbol correcting codes are well tailored. Thus, RS codes and their non-binary LDPC counterpart will be the basis for the single-level and two-level coding schemes used in this thesis.

In addition to electronic AWGN, the recording read channel is corrupted by data-dependent correlated additive Gaussian noise. This noise source model is dominated by jitter noise, which results from the variations of the location of bit transitions, where those variations are due to the zigzag-like boundaries separating the magnetization domains composing the media. To account for this additional noise source, data-pattern dependent noise prediction (PDNP) is incorporated in the channel detector matched to the ISI channel.

An approximate read channel model that encompasses many of the properties men-

tioned here is the first-order jitter noise model [58], shown in Figure 3.1. In this model, $s(t)$ is the dibit response, which is related to the transition response, $h(t)$, by:

$$s(t) = \frac{1}{2}\left[h(t) - h(t-T)\right]$$

where T is the symbol period. The PMRC transition response can be modelled by the hyperbolic tangent continuous waveform [59], which is given by:

$$h(t) = \tanh\left(\frac{2t}{0.5795 \times \pi \times pw_{50}}\right) \tag{3.1}$$

where pw_{50} is the width of the transition response defined as the time interval between the points at -50% and 50% of the saturation level. A measure of how packed the coded bits are on the media per unit area is the normalized channel density, defined as:

$$D_s \equiv pw_{50}/T. \tag{3.2}$$

This quantity is related the normalized user data density, denoted by D_u, by the relation:

$$D_u = D_s \times R. \tag{3.3}$$

where R is the code rate, and this relation summarizes the penalty of using more redundancy to guarantee more reliable recording, where to achieve the same user density D_u at a lower code rate R, the channel density D_s has to be proportionally increased. Hence, since pw_{50} is fixed (the recording channel is "transition response fixed"), a lower symbol period T is required, resulting in increased ISI.

As can be seen in Figure 3.1, two noise sources corrupt the data path: electronic AWGN noise $n(t)$, and the noise process $j(t)$, which models the position jitter of the transition pulse, and is given by:

$$j(t) = \sum_{k=-\infty}^{\infty} h'(t) * \left[\frac{j_k b(t-kT) - j_k b(t-(k-1)T)}{2}\right]. \tag{3.4}$$

where $h'(t)$ is the derivative transition response, $b(t)$ is the continuous waveform corresponding to the NRZ sequence, and j_k is an uncorrelated Gaussian-distributed sequence.

The jitter noise model of (3.4) summarizes the fact that more transitions in the recorded sequence inject more jitter-related noise power in the data path. The first stage of the analogue front end (AFE) is the 7th-order butterworth low-pass filter $p(t)$, which is there to prevent aliasing after sampling. Since this model assumes no high-pass filtering in the AFE, and since the dibit response is DC-full, a baseline wandering compensation loop is assumed in the data path to enforce dc-coupling [60].

After sampling of the analogue readback signal, the continuous channel model can be reduced to the discrete model shown in Figure 3.2, where:

$$s_k^s \equiv [s(t) * p(t)]_{t=kT}\,,\ h_k^j \equiv [h'(t) * p(t)]_{t=kT}\,,\ h_k^n * h_{-k}^n = [p(t) * p(-t)]_{t=kT}\,.$$

A useful definition of the system SNR that facilitates performance comparison of read channels that operate at different symbol densities, due to varying code rates, was proposed in [61]. Specifically, [61] proposes the following SNR definition:

$$SNR = \frac{E_{dt}}{N_o + M_o} = \frac{E_{dt}}{N_\alpha}. \tag{3.5}$$

where N_o is the electronic AWGN noise spectral height, M_o is twice the average energy of the medium noise associated with each transition, and E_{dt} is the average energy of $h'(t)$. Also, α, in $N_\alpha = N_o + M_o$, denotes the medium noise power expressed as a percentage of the total in-band noise power, and is given by:

$$\alpha = \frac{M_o}{M_o + N_o} \times 100\%. \tag{3.6}$$

As desired, [61] shows that (3.5) is free of channel density, provided that the intrinsic medium noise power per transition remains the same as the symbol pattern or density changes. Hence, using this definition of SNR makes it fare to compare systems running at different rates, and thus, this SNR expression will be adopted in this thesis.

The sampled low-pass filtered dibit response, s_k^s, can be quite long, especially for high density recording channels. This means that a trellis-based channel detector matched to s_k^s, such VA or BCJR, will be very complex to implement. The solution is to equalize s_k^s

to a short partial response target, and thus reduce the complexity of the matching channel detector in what is termed partial response maximum likelihood (PRML) detection [62–64]. In Figure 3.3 we show the dibit response for a channel of normalized density $D_s = 1.1$, and a noise mixture of 90% jitter noise and 10% electronic AWGN. A 2-tap PR target with low output noise correlation for the all-transition sequence is found to be $1 + 0.85D$. In Figure 3.3, we also show the frequency response of the dibit response and suggested PR target. The corresponding 15-tap FIR equalizer, found by least squares, is also shown. Moreover, the frequency responses of the equalized dibit and target are shown to match well for low frequency. While for high frequency the equalizer fails to enforce the PR target, resulting in unavoidable SNR loss due to the mismatch.

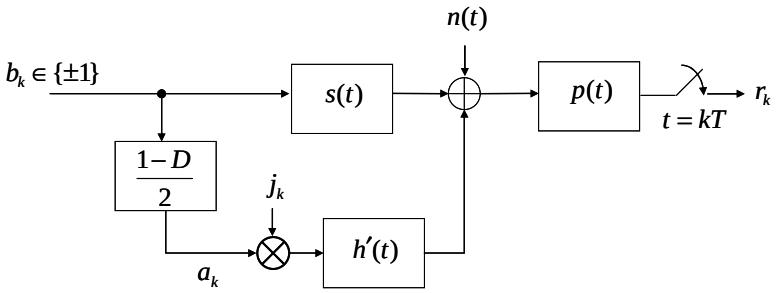


Figure 3.1: Continuous first-order jitter-noise model of PMRC.

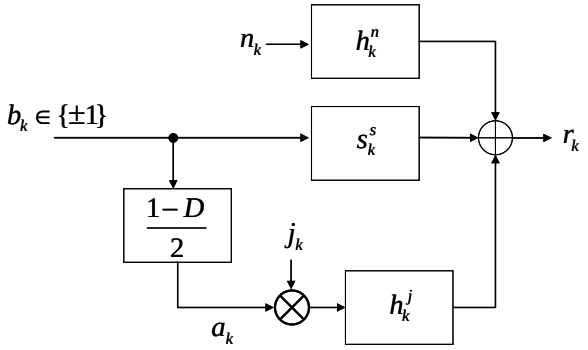


Figure 3.2: Discrete first-order jitter-noise model of PMRC.

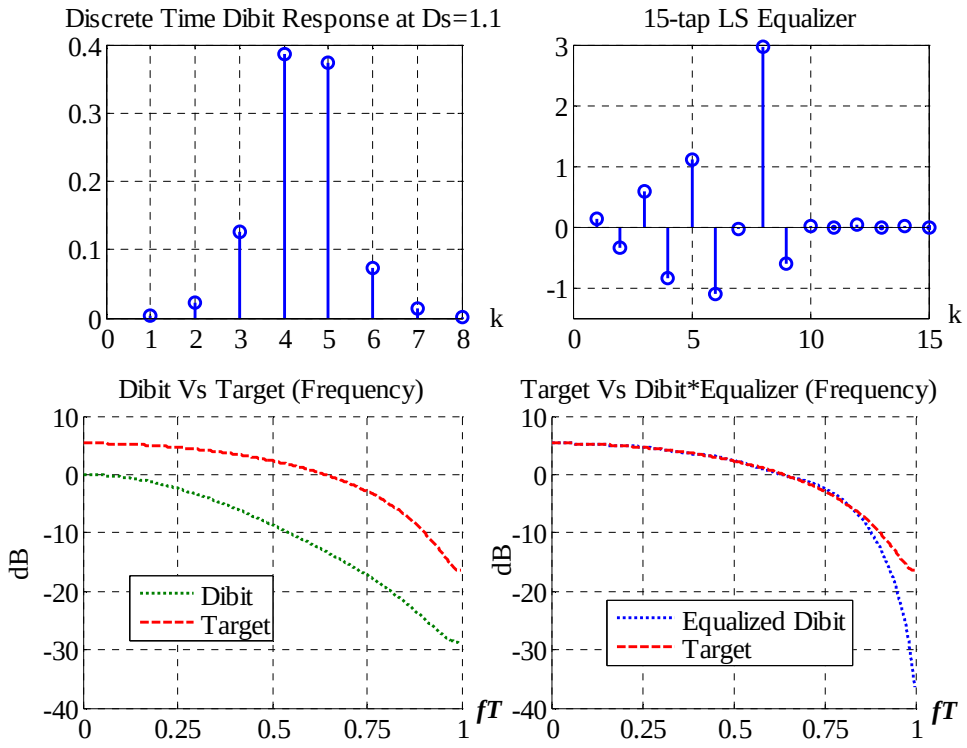


Figure 3.3: PRML equalization of the sampled dibit response to a short PR target.

3.2 Noise-Predictive Patterned-Matched Correlators

Error pattern reliability measures are computed by the maximum *a posteriori* (MAP)-based error-pattern correlator shown in Figure 3.4. The correlator design discussed here is an extension of the AWGN-environment correlator presented in [40] to the media noise dominated environment, in which pattern-dependent noise prediction is incorporated in the sliding-window correlator metric. The modified pattern-dependent correlator scans the ML word, estimating the data-dependant noise in the process without requiring any trellis expansion.

Let r_k be the channel detector input sequence $r_k = c_k * h_k + n_k^e + n_k^j(c_k)$, where c_k is the bipolar representation of the transmitted codeword sequence, h_k is the partial response target of length l_h, n_k^e is the zero-mean AWGN electronic noise with variance σ_e^2, and $n_k^j(c_k)$ is the zero-mean correlated data-dependant media noise with variance σ_j^2. Also, let

$q_k = r_k - (\hat{c}_k * h_k) = (c_k - \hat{c}_k) * h_k + n_k$ be the channel detector's output error sequence (with the channel detector utilizing PDNP via an expanded trellis [65]), and $n_k = n_k^e + n_k^j(c_k)$. If a target error pattern sequence $e_k^{(i)}$ occurs at positions from $k = j$ to $k = j + l_i - 1$, then this q_k can be written as

$$\begin{aligned} q_k &= (c_k - \hat{c}_k) * h_k + n_k \\ &= [\mathbf{e}^{(i)}]_j^{j+l_i-1} * h_k + n_k \\ &= [\mathbf{s}^{(i)}]_j^{j+l_i^h} + n_k \end{aligned} \tag{3.7}$$

where the notation $[\mathbf{x}]_i^j$ denotes a local segment $[x_i, x_{i+1}, ..., x_j]$ of the sequence x_k, $s_k^{(i)}$ is the noise-free error signal given by $s_k^{(i)} = e_k^{(i)} * h_k$, and $l_i^h = l_i + l_h - 2$.

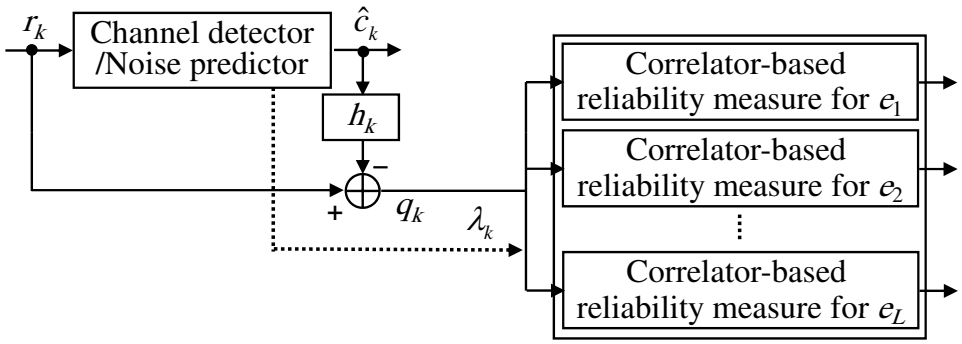


Figure 3.4: Local error-pattern correlators

The reliability measure for each error starting position $\rho_m^{(i)}$ can be computed by the local *a posteriori* probabilities

$$P\left([\mathbf{s}^{(i)}]_j^{j+l_i^h} \middle| [\mathbf{r}]_{j-l_p}^{j+l_i^h}, [\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\right) = P\left([\mathbf{s}^{(i)}]_j^{j+l_i^h} \middle| [\mathbf{q}]_{j-l_p}^{j+l_i^h}, [\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\right),$$

where $l^{p,h} = l_p + l_h - 1$, $j \in \{\rho_1^{(i)}, \rho_2^{(i)}, \cdots, \rho_M^{(i)}\}$, and l_p PDNP taps are employed. Using Bayes' theorem, the *a-posteriori* probability can be rewritten as

$$\begin{aligned} &P\left([\mathbf{s}^{(i)}]_j^{j+l_i^h} \middle| [\mathbf{q}]_{j-l_p}^{j+l_i^h}, [\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\right) \\ &= \frac{P\left([\mathbf{q}]_{j-l_p}^{j+l_i^h} \middle| [\mathbf{s}^{(i)}]_j^{j+l_i^h}, [\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\right) P\left([\mathbf{s}^{(i)}]_j^{j+l_i^h}\right)}{P\left([\mathbf{q}]_{j-l_p}^{j+l_i^h} \middle| [\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\right)}. \end{aligned} \tag{3.8}$$

Here, $P\Big([\mathbf{q}]_{j-l_p}^{j+l_i^h}\Big|[\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\Big)$ can be approximated by

$$\begin{aligned}
&P\Big([\mathbf{q}]_{j-l_p}^{j+l_i^h}\Big|[\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\Big) \\
&\approx P\Big([\mathbf{q}]_{j-l_p}^{j+l_i^h}\Big|[\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h},[\mathbf{s}^{(i)}]_{j}^{j+l_i^h}\Big)\cdot P\Big([\mathbf{s}^{(i)}]_{j}^{j+l_i^h}\Big) \\
&+P\Big([\mathbf{q}]_{j-l_p}^{j+l_i^h}\Big|[\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h},[\tilde{\mathbf{s}}^{(i)}]_{j}^{j+l_i^h}\Big)\cdot P\Big([\tilde{\mathbf{s}}^{(i)}]_{j}^{j+l_i^h}\Big)
\end{aligned} \tag{3.9}$$

where $[\tilde{\mathbf{s}}^{(i)}]_{j}^{j+l_i^h}$ corresponds to the most probable competing error pattern, and by definition, this is the all-zero error pattern with respect to the ML detector's decision sequence (i.e., the most probable competing local pattern is that associated with the ML word itself). Examining (3.7), and modelling the media noise using an l_p-tap PDNP, q_k can be represented as a sequence of statistically independent Gaussian random variables with mean $\hat{n}_k^p([\hat{\mathbf{c}}]_{k-l^{p,h}}^{k})$ and variance $\sigma_k^2([\hat{\mathbf{c}}]_{k-l^{p,h}}^{k})$, which reflect the predictor noise estimate and noise prediction error variance, respectively. Thus, the likelihood probabilities in the RHS of (3.9) are given by (with sequence upper and lower indices dropped for decluttering the notations)

$$\begin{aligned}
P\big(\mathbf{q}\big|\mathbf{s}^{(i)},\hat{\mathbf{c}}\big) &= \frac{(2\pi)^{-l_i^h/2}e^{-\sum_{k=j}^{j+l_i^h}\frac{(q_k-s_k^{(i)}-\hat{n}_k^p(-\hat{\mathbf{c}}_k))^2}{2\sigma_k^2(-\hat{\mathbf{c}}_k)}}}{\prod_{k=j}^{j+l_i^h}\sqrt{\sigma_k^2(-\hat{\mathbf{c}}_k)}} \\
P\Big(\mathbf{q}\Big|\tilde{\mathbf{s}}^{(i)},\hat{\mathbf{c}}\Big) &= \frac{(2\pi)^{-l_i^h/2}\,e^{-\sum_{k=j}^{j+l_i^h}\frac{(q_k-\hat{n}_k^p(\hat{\mathbf{c}}_k))^2}{2\sigma_k^2(\hat{\mathbf{c}}_k)}}}{\prod_{k=j}^{j+l_i^h}\sqrt{\sigma_k^2(\hat{\mathbf{c}}_k)}}.
\end{aligned} \tag{3.10}$$

The two *a priori* probability terms in the RHS of (3.9) are given by

$$\begin{aligned}
P\Big([\mathbf{s}^{(i)}]_{j}^{j+l_i^h}\Big) &= \prod_{k=0}^{l_i-1}P\left(c_{j+k}=(-1)^{|\frac{e_k^{(i)}}{2}|}\hat{c}_{j+k}\right) \\
P\Big([\tilde{\mathbf{s}}^{(i)}]_{j}^{j+l_i^h}\Big) &= \prod_{k=0}^{l_i-1}P\left(c_{j+k}=\hat{c}_{j+k}\right).
\end{aligned} \tag{3.11}$$

For an equally-probable c_k, i.e., $P(c_k=\pm1)=\frac{1}{2}$, we have $P([\mathbf{s}^{(i)}]_{j}^{j+l_i^h})=P([\tilde{\mathbf{s}}^{(i)}]_{j}^{j+l_i^h})$. However, if *a priori* information λ_k is available through another detector/decoder stage so that

$$P(c_k=+1)=\frac{e^{\lambda_k}}{1+e^{\lambda_k}}\quad\text{, and }\quad P(c_k=-1)=\frac{1}{1+e^{\lambda_k}}, \tag{3.12}$$

where $\lambda_k = \log \frac{P(c_k=+1)}{P(c_k=-1)}$, then $P([\mathbf{s}^{(i)}]_j^{j+l_i^h}) \neq P([\tilde{\mathbf{s}}^{(i)}]_j^{j+l_i^h})$.

Utilizing (3.10) and (3.9), taking the log-ratio between the *a posteriori* probability (3.8) and its counterpart *a posteriori* probability for $[\tilde{\mathbf{s}}^{(i)}]_j^{j+l_i^h}$ gives

$$C\left(e_j^{(i)}\right) = \sum_{k=j}^{j+l_i^h} \frac{1}{2}\Big[\frac{(q_k - \hat{n}_k^p(\hat{\mathbf{c}}_k))^2}{\sigma_k^2(\hat{\mathbf{c}}_k)} - \frac{(q_k - s_k^{(i)} - \hat{n}_k^p(-\hat{\mathbf{c}}_k))^2}{\sigma_k^2(-\hat{\mathbf{c}}_k)} + \log\frac{\sigma_k^2(\hat{\mathbf{c}}_k)}{\sigma_k^2(-\hat{\mathbf{c}}_k)}\Big] - \log\frac{P([\tilde{s}^{(i)}]_j^{j+l_i^h})}{P([s^{(i)}]_j^{j+l_i^h})}. \tag{3.13}$$

Equation (3.13) represents the "local" error-pattern-dependent noise-predictive correlator output in the sense that it essentially describes the correlator operation between q_k and the channel output version of the dominant error pattern $e_j^{(i)}$ within the local region $[j, j + l_i^h]$, while accounting for the data-dependant correlated noise and any available side information.

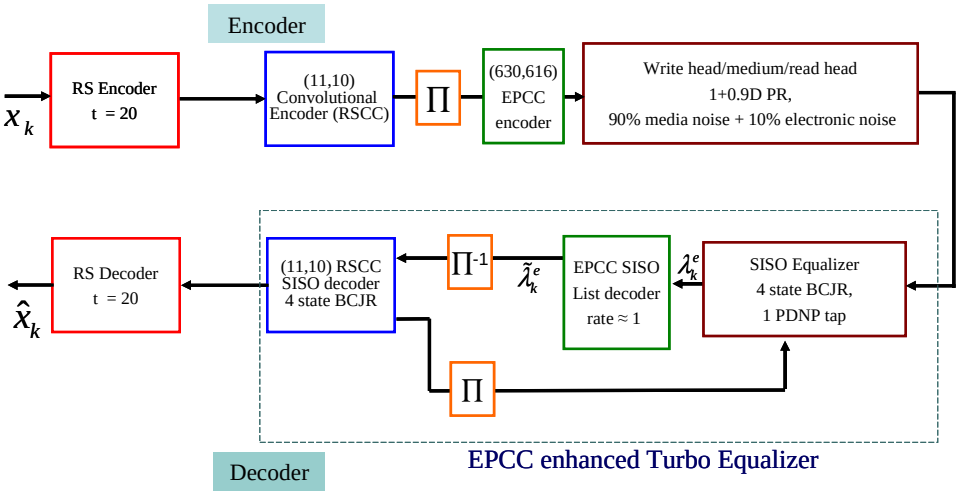


Figure 3.5: TE-EPCC block diagram for application in PMRC.

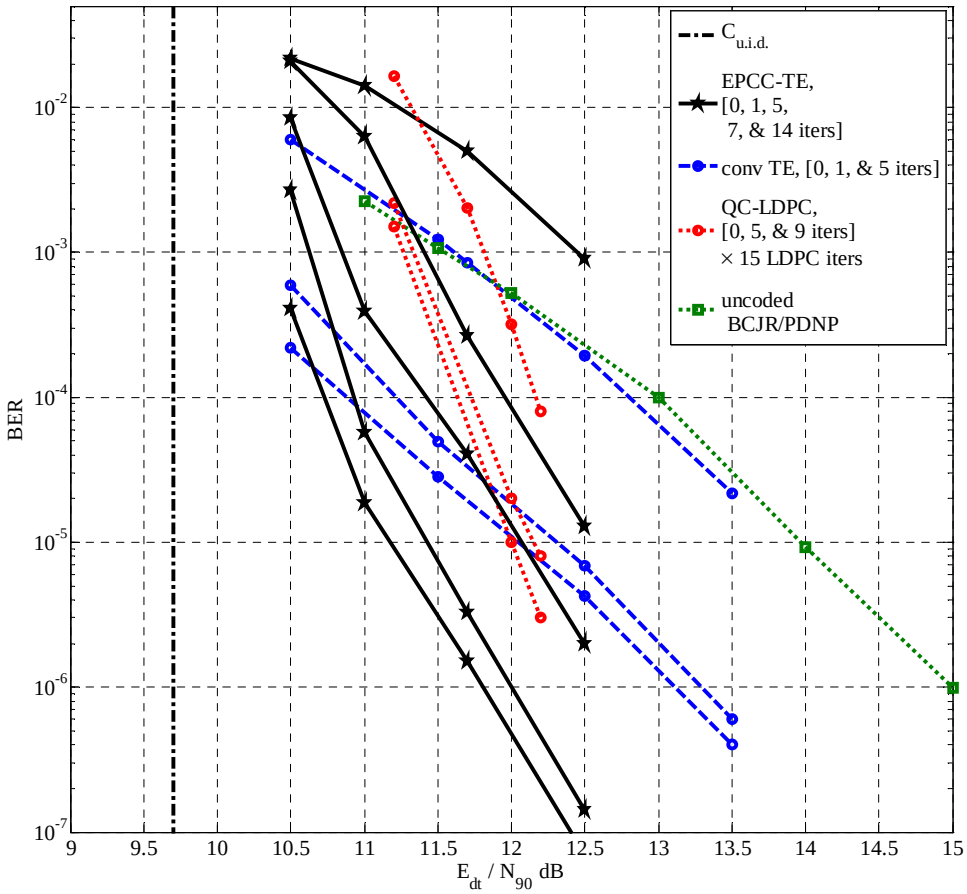


Figure 3.6: BER comparison of conventional TE, TE-EPCC, QC-LDPC, and uncoded BCJR/PDNP, compared to i.u.d. capacity at a user density of 1.2, different channel densities, and 90% media noise.

3.3 Performance Evaluation of TE-EPCC for PMRC

We investigate the performance of the proposed TE-EPCC scheme in a perpendicular magnetic recording channel (PMR) environment. The PMR read channel is modelled by a hyperbolic tangent transition response with normalized channel density D_s, with rate penalty proportional to $\frac{1}{R^2}$, measured in dB scale, for a system running at a rate R. The normalized channel density is defined as the ratio of the width over -50% to 50% of the transition response's saturation level to the user bit period. The PMR channel is equalized to the partial

response target $1 + 0.9D$, which was found to be among the best targets for the indicated environment that minimize the number of PDNP taps required to model the data-dependent noise memory. The noise mixture is assumed to be 10% AWGN and 90% jitter noise. For these channel parameters, the 10 most dominant error events are the polarity alternating error sequences of length 1 to 10 of the form $\pm[2, -2, 2, -2, ...]$. The SNR has been defined as the energy of the first derivative of the transition response E_{dt} to the noise spectral density N_{90}, which corresponds to 90% jitter noise. The channel detector is BCJR with 1 PDNP tap running on an expanded trellis of 4 states.

Conventional TE

An outer 4-state $(7, 5)_{oct}$ RSC code having punctured rate $\frac{8}{9}$ is serially concatenated to the channel through a 4401 bit interleaver. The normalized user density is $D_u = 1.2$, and the normalized channel density is $D_s = 1.35$.

TE-EPCC

The TE-EPCC operates at the same rate as the conventional TE but divides the correction power between a punctured $\frac{10}{11}$ outer $(7, 5)_{oct}$ RSC and an inner $\frac{616}{630}$ EPCC. The encoded output of the outer RSC is bit-interleaved by a 4401 bit interleaver and encoded into seven $(630, 616)$ EPCC codewords forming one sector that is passed to the channel without interleaving as shown in Figure 3.5. The normalized user density is again $D_u = 1.2$, and the normalized channel density for TE-EPCC is also $D_s = 1.35$. The pattern-dependant correlator incorporates 4 PDNP taps, without requiring trellis expansion, to estimate the media noise given the input hard ML codeword, where the ML codeword is the output of the BCJR channel detector employing 1 PDNP tap. A total of 25 test words were constructed, each containing up to 2 corruptions, which allows the correction of up to $m_c = 3$ multiple occurrences. In the soft output generation stage of the EPCC SISO decoder, the value of the back-off factor was increased gradually each turbo iteration

as $\beta^{iter} = [0.4, 0.4, 0.5, 0.6, 0.6, 0.8, 0.8, 0.9, 0.9, 1, 1, 1, 1, 1, 1]$, and λ_{max} in (2.2) was increased from 7 to 15 as SNR increased. Details on the design of the EPCC single pattern correcting code can be found in [38].

Quasi-Cyclic LDPC

The benchmark LDPC code was designed using the algebraic structured technique in [66], [67] to have performance that rivals the best comparable random codes at the design block length of 4699 and $R = 0.81$ with considerably lower encoding complexity. The LDPC parity check matrix is made up of a 7×37 array of circulant submatrices, where the $(i, j)^{th}$ submatrix is a 127×127 identity matrix with its rows cyclically shifted to the left by $(18^i \times 16^j) \mod 127$. The constructed LDPC code as such is a $(7, 37)$ regular code, that is quasi-cyclic (QC) with period 37. In the decoding process of this QC-LDPC code, 15 LDPC message-passing decoder iterations were performed each channel iteration.

Channel Capacity

To have an insight into the significance of the performance gap between the compared systems, their BER performance is compared to the capacity of the PMR channel constrained to have independent uniformly distributed (i.u.d.) binary inputs, where the i.u.d. capacity is identical to the channel capacity at high code rates. The i.u.d. capacity of a media noise dominated channel is evaluated numerically exploiting the forward recursion of the BCJR/PDNP algorithm [68]. For a PMR channel with normalized channel density $D_s = 1.48$, and 90% media noise, the minimum SNR required to achieve a reliable rate $R = 0.81$ was computed to be 9.7 dB.

The BER comparison is shown in Figure 3.6. Since it is hard to estimate the error rate performance involving an outer RS code, the BERs of TE systems were simply simulated for $R = \frac{8}{9}$ and $D_s = 1.35$ without RS coding, although we envision TEs to be used with RS outer coding in practice. The actual gain for using an additional RS code of, say, $t = 20$

(lowering the overall code rate to 0.81) would depend on its correction capability in the presence of certain per-sector probability distribution of symbol errors, after considering the rate penalty associated with it. For the outer $t = 20$ RS code of rate $R_{rs} = 0.91$, the rate penalty is reasonably estimated as $10 \times log_{10}(\frac{1}{R_{rs}^2}) = 0.82$ dB.

Simulations have shown that the BER of conventional TE saturates by the second iteration while the TE-EPCC performance gradually improves, and does not show a sign of saturation before the 9-th iteration. Figure 3.6 shows that TE-EPCC with 25 test error words has a gain of 0.8 dB compared to conventional TE at a target BER of 10^{-5}. Moreover, in the worst case scenario of an outer $t = 20$ RS offering no coding gain after rate penalty, TE-EPCC would still be as good as QC-LDPC, and is 2.3 dB away from i.u.d. capacity. The TE-EPCC BER gains in this low-to-medium SNR region are beneficial if the per-sector symbol error burst statistics are not severe, since then, an outer RS code can drive the systems sector error rate (SER) down to a satisfactory low SER required in commercial disk drives. Running a Monte Carlo simulation of the per-sector symbol error probability distribution of TE-EPCC, we could observe that TE-EPCC does exhibit somewhat worse error propagation over the conventional TE system. This additional burstiness can be attributed in part to errors in the unprotected parity bits of EPCC teaming up with errors elsewhere to form multiple error occurrences that resist correction, with interleaving making the situation worse by spreading bit errors over several output symbol errors. To clarify further, for the inner EPCC parity, no *a-priori* information is supplied by the outer convolutional code. Thus, the EPCC correction power for those bits does not improve with more channel iterations, as is observed during simulation. On the other hand, our simulations show that the featured QC-LDPC [66, 67] generates a large number of erroneous symbols per sector on the onset of a decoding failure. If this QC-LDPC is concatenated to an outer RS code, most failed sectors it generates cannot be handled by the outer RS for any reasonable symbol-correction power.

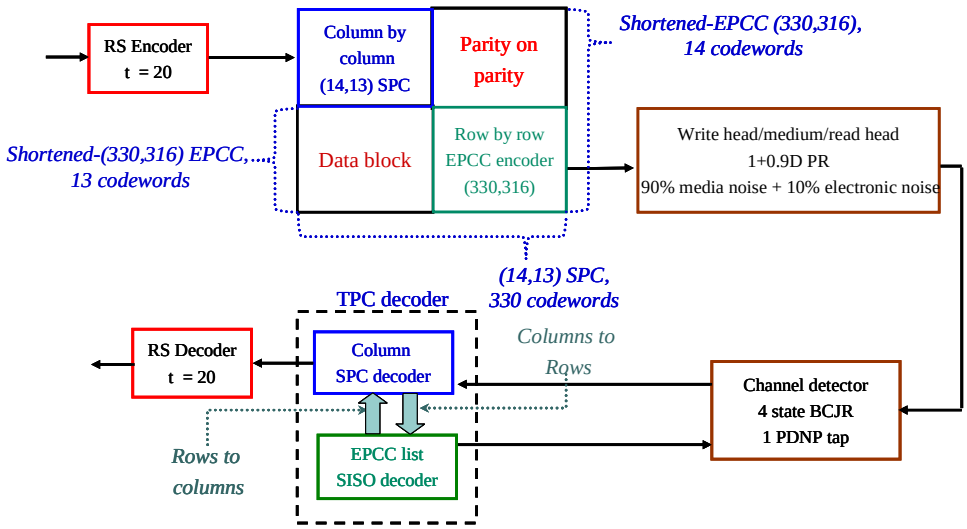


Figure 3.7: TP-EPCC block diagram for application in PMRC.

3.4 Turbo-Product EPCC for PMRC

One drawback of enhanced TE involving the serial concatenation of an interleaved RSCC and EPCC is that the parity bits introduced by the addition of the inner EPCC are not protected by the outer RSCC. Thus, the *a priori* information of these bits is not updated by turbo channel iterations. Although error events in those bits do not count towards the final BER, since EPCC parity bits get discarded prior to deinterleaving, they can still pair up with error events elsewhere in the codeword to form multiple-error-pattern occurrences that are not always resolvable by the EPCC-turbo action. Motivated by this design limitation, we introduce a turbo product code (TPC) design in which EPCC is the component row-encoder, termed TP-EPCC. In TP-EPCC, the parity bits of EPCC are protected by the column-wise single parity code (SPC), and at the same time, the column formed by the parity bits of SPC is also an EPCC codeword. This unique property means that the soft decoders matched to the component codes can exchange *a priori* information corresponding to each others parity bits.

In the design of the component codes, we found that a good approach to lower the block

length of EPCC to be incorporated in TP-EPCC is to use shortening of a longer-codeword EPCC. Shortening while maintaining the same redundancy results in a lower code rate and higher rate penalty. Still, maintaining the syndrome sets of the base long EPCC increases the accuracy of the single decoder. This is because a syndrome set period will decode to fewer possible error positions for the shorter codeword.

Another approach that we have investigated in designing a short codeword length EPCC is to target a smaller number of dominant error patterns, then, the generator polynomial degree will be less, and a higher rate EPCC code of short length might be easier to design. However, we found that TP-EPCC based on EPCC codes shortened from the long $(630, 616)$ EPCC deliver superior performance compared to codes derived directly from short base codes, such as $(210, 199)$ EPCC or $(186, 177)$ EPCC.

The TP-EPCC component codes are a $(330, 316)$ shortened EPCC of rate 0.96, derived from $(630, 616)$ EPCC, and the $(14, 13)$ single parity code (SPC) of rate 0.93. The total rate of TP-EPCC is the product of the component code rates, and is thus equal to 0.91. The TP-EPCC code block size is the product of the component codeword lengths given by 330×14. Interleaving is performed by mapping EPCC row-codewords to SPC column-codewords, and vise versa, which is sufficient to decorrelate the bit LLRs exchanged between the SISO decoders matched to the component codes.

In the simulation study of TP-EPCC, we have found that even with such a simple SPC as the column encoder, and the simple deterministic interleaving, TP-EPCC exhibits one order of magnitude better BER compared to a similar rate TE-EPCC based on a rate $\frac{10}{11}$ convolutional code, as will be shown in detail later in this section.

In the TP-EPCC soft iterative decoder, the column SPC code uses the sum of the channel detector and row-EPCC extrinsic output LLRs as its channel observation, then, the SPC output extrinsic LLR is passed as *a priori* information to the channel detector. Several such global iterations are performed before the TP-EPCC *posteriori* LLRs are sliced to generate output decisions. To achieve the best iterative performance, we have run 5 inner TP-EPCC

iterations and 15 channel iterations. Although the EPCC decoder is run more frequently in TP-EPCC compared to EPCC-TE, this would not result in worsened delay, if we unroll the iterative loop through hardware duplication.

A block diagram of TP-EPCC is shown in Figure 3.7. Based on the algebra of soft bits discussed in [69], the TP-EPCC iterative soft decoder flowchart can be designed as:

1. EPCC SISO decoder:
   - *A priori* soft input: map SPC extrinsic information λ_e^{spc} from columns to rows.
   - Initialization: use the channel detector extrinsic information plus SPC's extrinsic output, $\lambda_e^{ch} + \lambda_e^{spc}$, in the calculation of the correlator pattern-level reliability measures $C\left(e_j^{(i)}\right)$.
   - Decoding: perform EPCC list decoding for each row $1 \leq i \leq 14$, and use the list of candidate codewords to generate EPCC output extrinsic information λ_e^{epcc}.
2. SPC SISO decoder:
   - *A priori soft* input: map EPCC extrinsic information λ_e^{epcc} from rows to columns.
   - Decoding: apply the SPC parity check equation for each column,$1 \leq j \leq 330$:

$$\lambda_{e,i,j}^{spc}\big|_{i=1}^{14} = 2\tanh^{-1}\Big(\prod_{1\leq k\leq 14, k\neq i} \tanh\left[\frac{\lambda_{e,k,j}^{ch} + \lambda_{e,k,j}^{epcc}}{2}\right]\Big)$$

3. Soft output and decisions:
   - Soft output: map SPC extrinsic information λ_e^{spc} to channel detector *a priori* information λ_a^{ch}.
   - Hard decisions: slice TP-EPCC total extrinsic information $\lambda_e^{spc} + \lambda_e^{epcc}$ into hard decisions to be passed to outer ECC.

For the same channel parameters as in Figure 3.6, we simulated the bit error rate of the proposed TP-EPCC with the design parameters indicated above and compared it to conventional TE in Figure 3.8. Both of the EPCC-based turbo systems in Figure 3.6 and Figure 3.8 employ 25 test words (10 single + 9 double + 6 triple ML word-corruptions) and 4 noise prediction taps for the pattern-dependent correlator. The results show that TP-EPCC is 1.5 dB better than conventional TE at BER 10^{-6} for the same channel parameters mentioned above and 5 inner TPC iterations per each of the 15 global channel iterations. TP-EPCC is also 0.2 dB better than TE-EPCC and 1.5 dB away from the channel capacity at the same BER. Nevertheless, EPCC enhanced iterative systems are still more bursty than uncoded systems, where for most of the time, sectors are free of errors. Once in a while, however, a failed sector will incur a large number of errors that are more than the outer RS correction power. To combat these rare occurrences, several RS sectors can be interleaved before applying RS. Hence, if a sector is error prone, while all other sectors are free of error, the symbol errors are distributed among the sectors, and each sector's share of errors becomes within the correction power of RS.

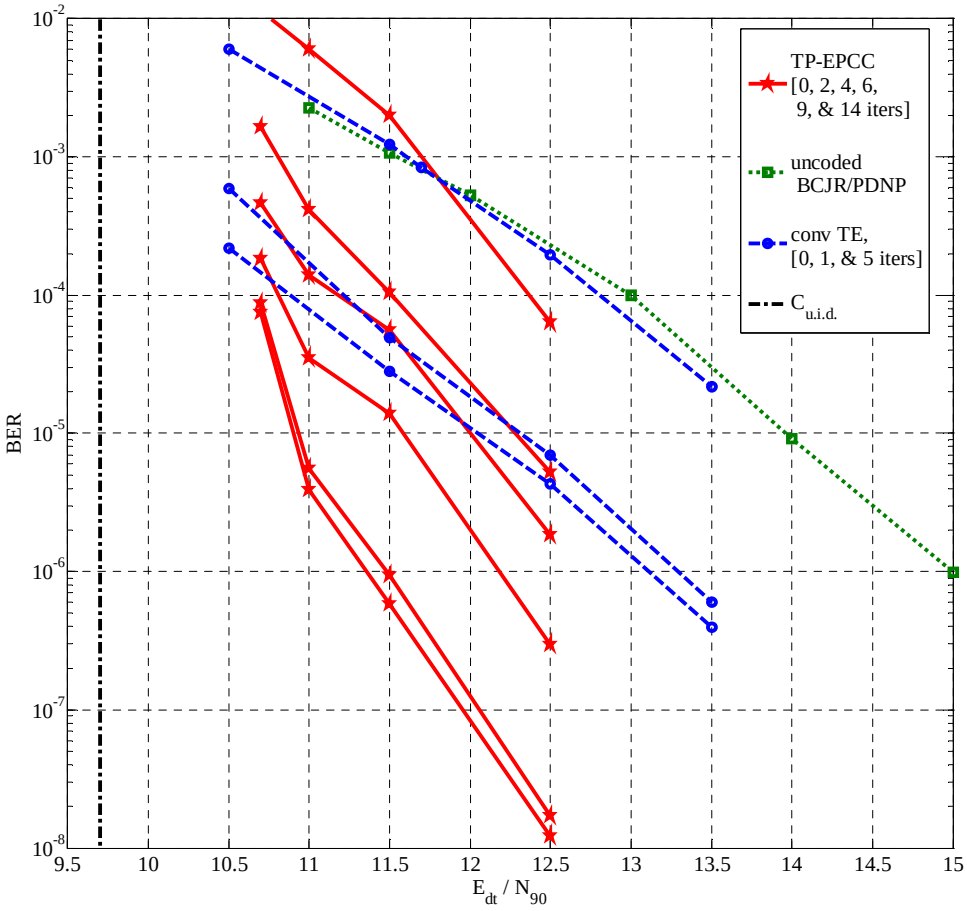


Figure 3.8: BER comparison of conventional TE, TP-EPCC, and uncoded BCJR/PDNP, compared to i.u.d. capacity at a user density of 1.2, different channel densities, and 90% media noise.

Chapter 4

Tensor Product Error Pattern Correcting Code (T-EPCC)

The error pattern correcting code (EPCC) can be constructed to provide a syndrome decoding table targeting the dominant error events of an inter-symbol interference channel at the output of the Viterbi detector. For the size of the syndrome table to be manageable and the list of possible error events to be reasonable in size, the codeword length of EPCC needs to be short enough. However, the rate of such a short length code will be too low for hard drive applications. To accommodate the required large redundancy, it is possible to record only a highly compressed function of the parity bits of EPCC's tensor product with a symbol correcting code. In this chapter, we show that the proposed tensor error-pattern correcting code (T-EPCC) is linear time encodable, show how to construct a two level T-RS-EPCC, and demonstrate analytically that tensor codes can potentially furnish gain for a wide range of coding rates and sector sizes provided that efficient decoding algorithms are feasible.

4.1 Tensor Product Parity Codes

We give a brief review of the tensor product coding paradigm and present an encoding method that allows linear-time-encodable TPPCs. Moreover, we present a novel fully analytic SER estimation method to evaluate the performance of ideal T-EPCC-RS.

4.1.1 Notations and Definitions

- For a ceratin parity check matrix H corresponding to a linear code $\{C : Hc^t = 0 \forall c \in C\}$, a syndrome s is the non-zero range of a perturbation of a codeword $H(c+e)^t = s$. A "virtual syndrome" refers to the range of H for any bit block, not necessarily formed of data and parity bits.

- For two matrices A and B, the tensor product $\otimes$ is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots \\ a_{21}B & a_{22}B & \\ \vdots & & \ddots \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{12}b_{11} & a_{12}b_{12} & \cdots \\ a_{11}b_{21} & a_{11}b_{22} & & a_{12}b_{21} & a_{12}b_{22} & \cdots \\ \vdots & & \ddots & & \vdots & \ddots \\ a_{21}b_{11} & a_{21}b_{12} & & & & \\ a_{21}b_{21} & a_{21}b_{22} & \cdots & & \ddots & \\ \vdots & \vdots & \ddots & & & \end{bmatrix} \tag{4.1}$$

4.1.2 Construction and Properties of the TPPC Parity Check Matrix

Consider a binary linear code $C_1 : (n_1, k_1, p_1)$ derived from the null space of parity check matrix H_1, and assume C_1 corrects any error event that belongs to class ε_1. Also, consider a non-binary linear code $C_2 : (n_2, k_2, p_2)$ derived from the null space of parity check matrix H_2 and defined over elements of $GF(2^{p_1})$. Moreover, assume this code corrects any symbol

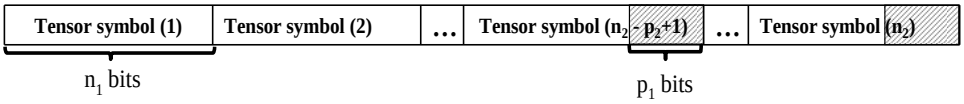


Figure 4.1: TPPC $C_1(n_1, k_1, p_1) \otimes C_2(n_2, k_2, p_2)$ codeword structure

error type that belongs to class ε_2. As a preliminary step, convert the binary $p_1 \times n_1$ matrix H_1, column by column, into a string of $GF(2^{p_1})$ elements of dimension $1 \times n_1$. Then, construct the matrix

$$H_3 = H_1 \otimes H_2$$

as a $p_2 \times n_1 n_2$ array of $GF(2^{p_1})$ elements. Finally, convert the elements of H_3 into p_1-bit columns, based on the same primitive polynomial of degree p_1 used all over in the construction method. The null space of the $p_1 p_2 \times n_1 n_2$ binary H_3 corresponds to a linear binary code $C_3 : (n_1 n_2, k_3, p_1 p_2)$. As shown in Figure 4.1, a C_3 codeword is composed of n_2 blocks termed "tensor-symbols", each having n_1 bits. Also, it can be shown that C_3 can correct any collection of tensor symbol errors belonging to class ε_2, provided that all errors within each tensor symbol belong to class ε_1 [43]. Note that a tensor symbol is not an actual C_1 codeword, and as such, using the terms "inner" and "outer" codes would not be completely accurate. In addition, the tensor symbols are not codewords themselves, as can be seen in Figure 4.1, the first k_2 tensor symbols are all data bits to start with, and even the last p_2 tensor symbols, which are composed of data and parity bits, have non-zero syndromes under H_1. Thus, a TPPC codeword does not correspond directly to either H_1 or H_2, and as a result, the codebooks they describe are not recorded directly on the channel.

Example TPPC, [43]

The simplest instance of EPCC is the class of single error correcting codes, the $(3, 1, 2)$ Hamming code being an example. The parity check matrix for this code is given by

$$H_{epcc} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \alpha & \alpha^2 \end{bmatrix}, \tag{4.2}$$

where we have also shown the $GF(2^2)$ representation of H_{epcc}. We will concatenate this double-parity code with the doubly-extended $t = 1$ $(5, 3)$ RS on $GF(2^2)$, which has the parity check matrix

$$H_{rs} = \begin{bmatrix} 1 & 0 & 1 & \alpha & \alpha^2 \\ 0 & 1 & 1 & \alpha^2 & \alpha \end{bmatrix}. \tag{4.3}$$

The tensor product of the two parity check matrices gives

$$H_{epcc} \otimes H_{rs} = \begin{bmatrix} 1 & \alpha & \alpha^2 & 0 & 0 & 0 & 1 & \alpha & \alpha^2 & \alpha & \alpha^2 & 1 & \alpha^2 & 1 & \alpha \\ 0 & 0 & 0 & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & \alpha^2 & 1 & \alpha & \alpha & \alpha^2 & 1 \end{bmatrix}. \tag{4.4}$$

Now, each Galois Field element is mapped back to its 2-bit representation. Hence, the parity check matrix of the TPPC code is:

$$H_{tppc} = \begin{bmatrix} 101 & 000 & 101 & 011 & 110 \\ 011 & 000 & 011 & 110 & 101 \\ 000 & 101 & 101 & 110 & 011 \\ 000 & 011 & 011 & 101 & 110 \end{bmatrix}. \tag{4.5}$$

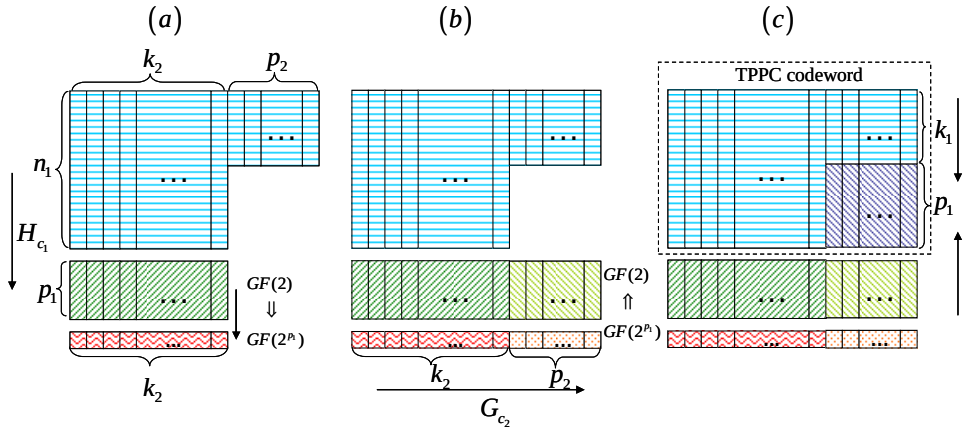


Figure 4.2: TPPC Encoder of $C_1(n_1, k_1, p_1) \otimes C_2(n_2, k_2, p_2)$:(a) virtual syndromes calculated for H_{c_1}, then the p_1-bit syndromes are mapped to $GF(2^{p_1})$ symbols, (b) k_2 virtual syndromes encoded by generator matrix G_2 into a C_2 codeword, then parity symbols are mapped to $GF(2)$, (c) $p_1 \times p_2$ TPPC parity bits are calculated by back substitution.

This is a binary $(15, 11)$ tensor product code that corrects any single tensor symbol error provided it contains a single bit error. Furthermore, although the EPCC constituent code has a very low rate of 0.34, the resulting TPPC has a high rate of 0.74. Moreover, the length of each tensor symbol is 3 bits, which has a lower probability of having more than one single bit error, given the significantly reduced rate penalty of the code, compared to stand-alone EPCC.

4.1.3 Encoding of Tensor Product Parity Codes

The encoding of TPPC can be performed using its binary parity check matrix, but the corresponding binary generator matrix is not guaranteed to possess algebraic properties that enable linear time encodability. Thus, an implementation-friendly approach would be to utilize the encoders of the constituent codes, which can be chosen to have linear time encoders.

Consider a binary code $C_1 : (n_1, k_1, p_1)$ thats is the null space of parity check matrix H_1, and a non-binary code $C_2 : (n_2, k_2, p_2)$ defined on $GF(2^{p_1})$, the tensor-product concatenation is a binary $C_3 : (n_3, k_3, p_3)$, where:

$$n_3 = n_2 \times n_1, k_3 = n_1 \times n_2 - p_1 \times p_2, p_3 = p_1 \times p_2$$

Assume that C_1 is a cyclic code, and C_2 is any of the linear time encodable codes, choosing quasi-cyclic (QC) for the purpose of this study. Then, the encoders of C_1 and C_2 communicate via the following algorithm to generate a codeword of C_3, see Figure 4.2:

(i) Receive a block of $n_1 \times k_2 + k_1 \times n_2 - k_1 \times k_2$ bits from the data source, call it major block α.

(ii) Divide major block α into minor block β of $n_1 \times k_2$ bits, and minor block γ of $k_1 \times n_2 - k_1 \times k_2$ bits (i.e. $k_1 \times p_2$ bits).

(iii) Divide block β into k_2 columns each of n_1 bits. Then, for each column, calculate the intermediate virtual p_1-bit syndrome under the parity check matrix of C_1. Using

a feedback shift register (FSR) to calculate the syndromes, the computational cost would be $O(n_1)$ operations per syndrome, and $O(n_1 \times k_2)$ for this entire step.

(iv) Convert intermediate virtual syndromes from p_1-bit strings into $GF(2^{p_1})$ symbols.

(v) Encode the k_2 non-binary syndromes into a C_2 codeword of length n_2. Using FSRs to encode the quasi-cyclic C_2, the computational complexity of this step would be $O(n_2)$.

(vi) Convert computed syndromes back into p_1-bit strings.

(vii) Divide block γ into p_2 columns each of k_1 bits. Add p_1 blanks in each column to be filled with the parity bits of C_3. Then, align each column with the p_2 syndromes computed in the previous step, leaving p_1 blanks in each column.

(viii) Fill blanks in the previous step such that the syndrome of data plus parity blanks under C_1 equals the corresponding aligned syndrome from step (vi). The parity can be calculated using the systematic H_1 and the method of back substitution which requires computational complexity $O(n_1)$ per column.

The total computational complexity of this encoding algorithm is $O(n_1 \times k_2 + n_2 + n_1 \times p_2 \sim n_1 \times n_2 = n_3)$. Thus, we have shown - with some constraints- that if C_1 and C_2 are linear time encodable, then $C_3 = C_1 \otimes C_2$ is linear time encodable!

4.2 T-EPCC-RS Codes

To demonstrate the algebraic properties of TPPC codes, we present an example code suitable for recording application with $1/2$K sector size. Consider two component codes:

- The binary cyclic $(18, 10)$ EPCC of example 2 above with rate 0.556, 8 parity bits,

and parity check matrix in $GF(2^8)$:

$$H_{epcc} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^{133} & \dots \\ & \alpha^{134} & \alpha^{96} & \alpha^{90} & \alpha^{82} & \alpha^{236} & \alpha^{234} & \alpha^{217} & \alpha^{92} & \alpha^{93} \end{bmatrix}_{1\times 18} \tag{4.6}$$

- A $(255, 195)$ RS over $GF(2^8)$, of rate 0.765, $t = 30$, and 60 parity symbols.

The resulting TPPC is a binary $(4590, 4110)$ code, of rate 0.895, and redundancy of 480 parity bits. For this code, a codeword is made of 255 18-bit tensor symbols, of which, any combination of 30 or less are correctable, provided that each 18-bit tensor symbol has a single or multiple occurrence of a dominant error that is correctable by EPCC, those being combinations of error polynomials up to degree 9. Furthermore, although the EPCC constituent code has a very low rate of 0.556, the resulting T-EPCC has a high rate of 0.895. Notably, in the view of the 18-bit EPCC, this 61% reduction in recorded redundancy corresponds to an SNR improvement of 2 dB in a channel with rate penalty $\sim 10\log_{10}(1/R)$, and 4.1 dB in a channel with rate penalty $\sim 10\log_{10}(1/R^2)$.

4.2.1 Hard Decoding of T-EPCC-RS Codes

Hard decoding of T-EPCC-RS directly reflects the code's algebraic properties and, thus, serves to further clarify the concept of tensor product codes. Hence, we discuss the hard decoding approach before going into the design of soft decoding of T-EPCC codes. The decoding algorithm is summarized by the following procedure (see Figure 4.3) :

(i) After hard slicing the output of the Viterbi channel detector, The virtual syndrome of each tensor symbol is calculated under H_{epcc}. Each virtual syndrome is then mapped into a Galois Field symbol, where the sequence of non-binary virtual syndromes constitute an RS codeword - if the channel detector did not suffer any errors.

(ii) Any hard-input RS decoder, such as the Berlekamp-Massey decoder [70], acts to find a legitimate RS codeword based on the observed syndrome-sequence.

(iii) If the number of syndrome-symbols in error is larger than the RS correction power, RS fails and the tensor product decoder halts.

(iv) Otherwise, if RS decoding is deemed successful, the corrected syndrome-symbol sequence is added to the original observed syndrome-symbol sequence to generate the "error syndrome-symbol" sequence.

(v) Each error syndrome-symbol is mapped into an EPCC bit-syndrome of the corresponding tensor symbol.

(vi) Finally, EPCC decodes each tensor symbol to satisfy the syndrome generated by the component RS, in which it faces two scenarios:

- A zero "error-syndrome" at the output of RS decoding indicates either no error occurred or a multiple error occurrence that has a zero EPCC-syndrome, which goes undetected. In this case the EPCC decoder is turned off to save power.
- A non zero "error-syndrome" will turn EPCC correction on. If the error-syndrome indicates a single error occurrence in the target set, then, the EPCC single error algebraic decoder is turned on. On the other hand, if the error-syndrome is not recognized, then EPCC list decoding is turned on with a reasonable-size list of test words.

Note that although the number of EPCC codewords (tensor symbols) is huge, the decoder complexity is reasonable since EPCC decoding is turned on only for nonzero syndromes.

4.3 RS Sector Error Rate Estimation

We present a fully analytic method for the estimation of the RS frame error rate in ISI channels. The method is inspired by the semi-analytical multinomial method [16, 20, 71]. Al-

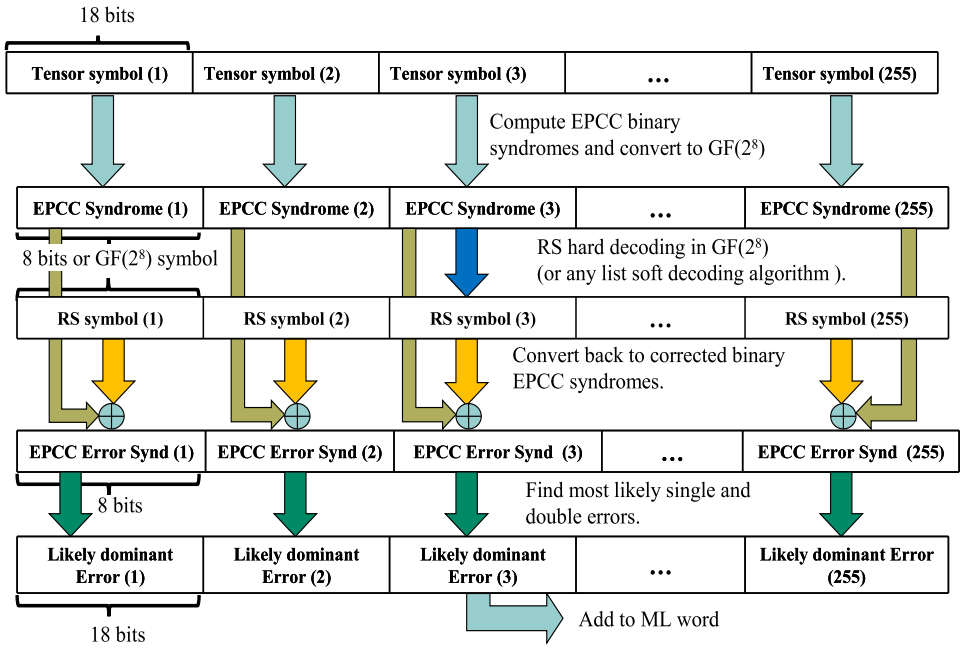


Figure 4.3: Hard decoder of $(18, 10, 8)$ EPCC $\otimes$ $(255, 195, 2t)$ RS of $t = 30$ over $GF(2^8)$.

though the block multinomial method is more accurate than the multinomial method [16], it is harder to analytically find the symbol error distribution per symbol block, which is necessary for block multinomial to find the SER of RS. On the other hand, it is easier to characterize the symbol error-event probabilities required by the multinomial method based on the known dominant error events at the output of Viterbi. We first present the classical simulation-based multinomial method and later show how we can avoid simulation altogether in the multinomial evaluation.

4.3.1 Semi-Analytic Multinomial Method

We assume that symbol error events at the output of Viterbi have a maximum length of m consecutive erroneous RS bytes, where a byte is a block of q bits. Also, assume that symbol error events are independent if separated by an error free interval of m bytes, which was observed to be less than 4 for our case study. Symbol error events are then characterized

by the joint probabilities $P(k,s)$, where $P(k,s)$ is the probability of an error event starting at symbol position k spanning s RS symbols. In simulation, these probabilities are found by sliding a window of size m symbols over the hard output of Viterbi and counting occurrences of 1 to m consecutive symbol errors. Since the probability of a symbol error event of any length is independent of the error location k, ignoring boundary effects, we need only find $P_s = \sum_k P(k,s)$. Then, it follows that the number of such symbol error events in an RS sector follows a multinomial distribution. RS fails if the number of symbol errors in one sector is larger than the RS correction power t. Hence, the probability of RS codeword failure, P_W, is the probability of $t+1$ or more errors in a sector of length n symbols, which is bounded from below by

$$\begin{aligned} P_W &\geq 1 - \sum_{s_0}\sum_{s_1}\cdots\sum_{s_m} \frac{n!}{s_0!s_1!\ldots s_m!} P_0^{s_0} P_1^{s_1}\ldots P_m^{s_m} \\ \forall s_i &: \sum_{i=0}^{m} i s_i \leq t, \sum_{i=0}^{m} s_i = n; \\ P_0 &= 1 - \sum_{i=1}^{m} P_i. \end{aligned} \tag{4.7}$$

This is a lower bound on the probability of m multi-symbol error events, with each symbol error event of length i occurring s_i times, totaling more than t symbol errors per sector. This bound was found to be tight for low SNR, but loosens as SNR increases. Next, if we can evaluate P_s, $s = 1,\ldots,m$, analytically, then, the entire multinomial method will be simulation free. This is discussed in the following section.

4.3.2 Analytic Multinomial Method

The output errors of Viterbi detection are dominated by a class of error patterns characterized by small diverging path energies. Let the truncated size of this error class be L. Also, let an error pattern from this class be $e^{(i)}, 1 \leq i \leq L$, that of which generates error event $e_k^{(i)}$ if it occurs at location k, $0 \leq k \leq nq$. We assume that errors at the output of channel detection come only from this list. Then, the probability of a symbol error starting

at symbol position l of length s is the sum of all error patterns that occur within symbol l, i.e. $l = k \bmod q$, and span s symbols. Furthermore, including error pattern combinations in addition to single occurrences increases estimation accuracy, and the number of such combinations is greatly reduced due to the constraint that the first symbol must be in error combined with the constraint that symbol errors are consecutive. Nevertheless, the same cannot be said for symbol error enumeration per block in the block multinomial method, in which symbol errors occur in all possible combinations. Denote the probability of $e_k^{(i)}$ by $p_k^{(i)}$, then, we have [72]:

$$P_s = \sum_{l=0}^{q-1} \sum_{k=(s-1)q-l+1}^{sq-l} p_l^{(k)}. \tag{4.8}$$

We note in general that $p_l^{(k)}$ are equal $\forall l$, i.e. error pattern probabilities are not function of the position of occurrence. Consider an ISI channel with dominant errors of the form: $e^{(1)}(x) = 1$, $e^{(2)}(x) = 1 + x$, $e^{(3)}(x) = 1 + x + x^2$, etc., i.e. polynomials over $GF(2)$ for which all powers of x have nonzero coefficients. This class of dominant errors encompasses a wide family of channel responses of interest in the magnetic recording industry. For this channel, the probability of a single symbol error, considering only single error-pattern occurrences, is given by:

$$P_1 = \sum_{k=1}^{q} (q - k + 1) p^{(k)} \tag{4.9}$$

whereas the probability of a double symbol error event is:

$$P_2 = \sum_{k=2}^{q} (k - 1) p^{(k)} + \sum_{k=1}^{q} (q - k + 1) p^{(q+k)} \tag{4.10}$$

and the probability of a triple symbol error event is:

$$P_3 = \sum_{k=1}^{q-1} k p^{(q+k+1)} + \sum_{k=1}^{q} (q - k + 1) p^{(2q+k)}. \tag{4.11}$$

Beyond symbol error event of length 3, errors are due more to multiple occurrences of patterns rather than long single error patterns. While it is hard to account for multiple error occurrences within the same symbol using this formulation, we can still combine error

patterns occurring in disjoint symbols that result in consecutive symbol error events. To approximate the effect of those multiple occurrences of error events we modify (4.10) and (4.11) to

$$\begin{aligned}
P_2 &= \sum_{k=2}^{q}(k-1)p^{(k)} + \sum_{k=1}^{q}(q-k+1)p^{(q+k)} \\
&\quad + \sum_{k_1=1}^{q}\sum_{k_2=1}^{q}(q-k_1+1)(q-k_2+1)p^{(k_1)}p^{(k_2)}. \\
P_3 &= \sum_{k=1}^{q-1}kp^{(q+k+1)} + \sum_{k=1}^{q}(q-k+1)p^{(2q+k)} \\
&\quad + 2\sum_{k_1=1}^{q}\sum_{k_2=2}^{q}(k_2-1)(q-k_1+1)p^{(k_2)}p^{(k_1)} \\
&\quad + 2\sum_{k_1=1}^{q}\sum_{k_2=1}^{q}(q-k_2+1)(q-k_1+1)p^{(q+k_2)}p^{(k_1)} \\
&\quad + \sum_{k_1=1}^{q}\sum_{k_2=1}^{q}\sum_{k_3=1}^{q}(q-k_1+1)(q-k_2+1)(q-k_3+1)p^{(k_1)}p^{(k_2)}p^{(k_3)}. \quad (4.12)
\end{aligned}$$

The probability of an error event $e^{(i)}$ of length l_i was shown in [40] to be

$$P(e^{(i)}) = \frac{1}{2^{l_i-1}}Q(\frac{\sqrt{E^{(i)}}}{2\sigma}). \quad (4.13)$$

where $E^{(i)}$ is the energy of the error event, and σ is the standard deviation of the AWGN noise process.

Within this family of channels, lies a subclass for which the energies of dominant error events are mostly equal, with shorter errors being more probable. Those are channels for which the metric difference between the correct and error paths are zero for path sections other than the diverging and merging branches. For example, the dominant errors of the partial response target (PR1) of channel response $1 + D$ have all energy $E^{(i)} = 8$. For this channel, we drop the index i and denote the energy for all dominant error events by E. In this case (4.9), and (4.12), which include the multiple-error correction factor, are simplified to:

$$
\begin{aligned}
P_1 &= (\frac{1}{2^{q-1}} + 2q - 2)Q(\frac{\sqrt{E}}{2\sigma}) \\
P_2 &= (\frac{1}{2^{2q-1}} - \frac{1}{2^{q-2}} + 2)Q(\frac{\sqrt{E}}{2\sigma}) + (\frac{1}{2^{q-1}} + 2q - 2)^2 Q^2(\frac{\sqrt{E}}{2\sigma}) \\
P_3 &= (\frac{1}{2^{3q-1}} - \frac{1}{2^{2q-2}} + \frac{1}{2^{q-1}})Q(\frac{\sqrt{E}}{2\sigma}) \\
&\quad +2(\frac{1}{2^{q-1}} + 2q - 2)(\frac{1}{2^{2q-1}} - \frac{1}{2^{q-2}} + 2)Q^2(\frac{\sqrt{E}}{2\sigma}) \\
&\quad +(\frac{1}{2^{q-1}} + 2q - 2)^3 Q^3(\frac{\sqrt{E}}{2\sigma}).
\end{aligned} \tag{4.14}
$$

Using the probabilities in (4.14), or the general case of (4.12), the analytic multinomial method estimates the SER to be:

$$
P_W \geq 1 - \sum_{s_0=0}^{n} \sum_{s_1=0}^{n-s_0} \sum_{s_2=0}^{n-s_0-s_1} \sum_{\substack{s_3=0 \\ s_1+2s_2+3s_3 \leq t}}^{n-s_0-s_1-s_2} \frac{n!}{s_0!s_1!s_2!s_3!} P_1^{s_1} P_2^{s_2} P_3^{s_3} (1 - P_1 - P_2 - P_3)^{s_0} \tag{4.15}
$$

Since the numbers involved in this calculation are very small numbers raised to large integers, this calculation can be performed in a finite precision machine only in log domain. For that end, we define the operators max_+^* and max_-^*. Assume $\alpha_L = \log(\alpha)$ and $\beta_L = \log(\beta)$, then:

- $(\alpha+\beta)_L = \log(e^{\alpha_L} + e^{\beta_L})$, and $max_+^*(\alpha_L, \beta_L) = (\alpha+\beta)_L = \max(\alpha_L, \beta_L) + \log(1 + e^{-|\alpha_L - \beta_L|})$.
- $(\alpha - \beta)_L = \log(e^{\alpha_L} - e^{\beta_L})$, and $max_-^*(\alpha_L, \beta_L) = (\alpha - \beta)_L = \max(\alpha_L, \beta_L) + \log(1 - e^{-|\alpha_L - \beta_L|})$.

then

$$
\begin{aligned}
\tilde{P}_W \geq\ & max_-^*(0, \max_{\substack{s_1+2s_2+3s_3 \leq t, s_0+s_1+s_2+s_3=n}}^{*} (\ln\Gamma(n+1) - \\
& \ln\Gamma(s_0+) - \ln\Gamma(s_1+1) - \ln\Gamma(s_2+1) - \ln\Gamma(s_3+1) \\
& +s_1\tilde{P}_1 + s_2\tilde{P}_2 + s_3\tilde{P}_3 + s_0 \times max_-^*(0, max_+^*(\tilde{P}_1, \tilde{P}_2, \tilde{P}_3)))
\end{aligned} \tag{4.16}
$$

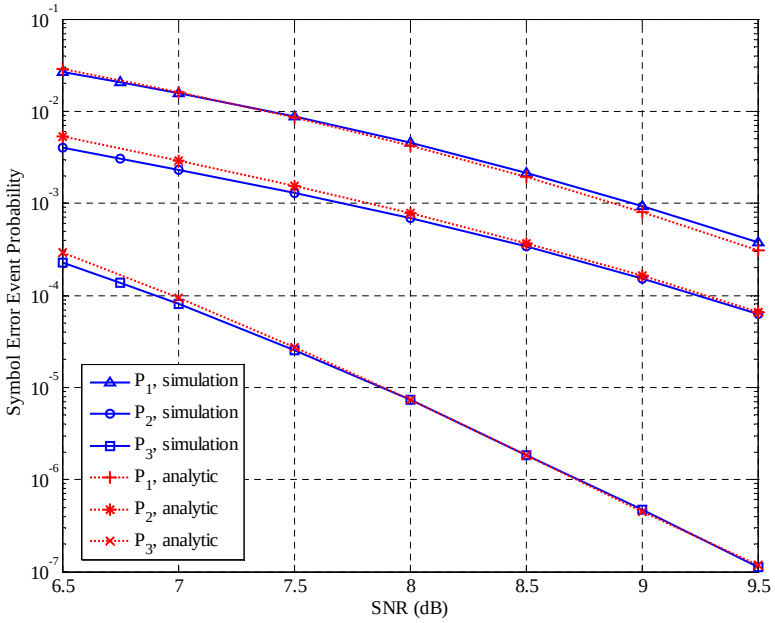


Figure 4.4: Comparison of simulation and analytic computation of symbol error event probabilities for $5 + 6D - D^3$ in AWGN, and 10-bit long symbols.

where $\tilde{x}$ is the log domain version of a variable x, and $\ln \Gamma(n) = \log([n-1]!)$ is the tabulated log gamma function.

4.4 Numerical Analytic and Simulation Results

We aim to study the performance advantage of the component RS in the tensor product of RS and EPCC, termed TP-RS, and single-level RS. we first validate the analytic estimation method above by count-based Monte Carlo simulations. We compare system performance in the ISI channel $5+6D-D^3$ corrupted by AWGN, which is a good PR target for perpendicular magnetic recording channels. For the first case study, we compare the performance of shortened $(450, 450-2t, 2t)$ RS over $GF(2^{10})$, and an equal rate TP-RS. The TP-RS is composed of an $(18, 10)$ EPCC and a shortened $(250, 250-2t_{tp})$ over $GF(2^8)$. Figure 4.4 shows the symbol error event probabilities of lengths 1, 2, and 3 for a 10-bit symbol based

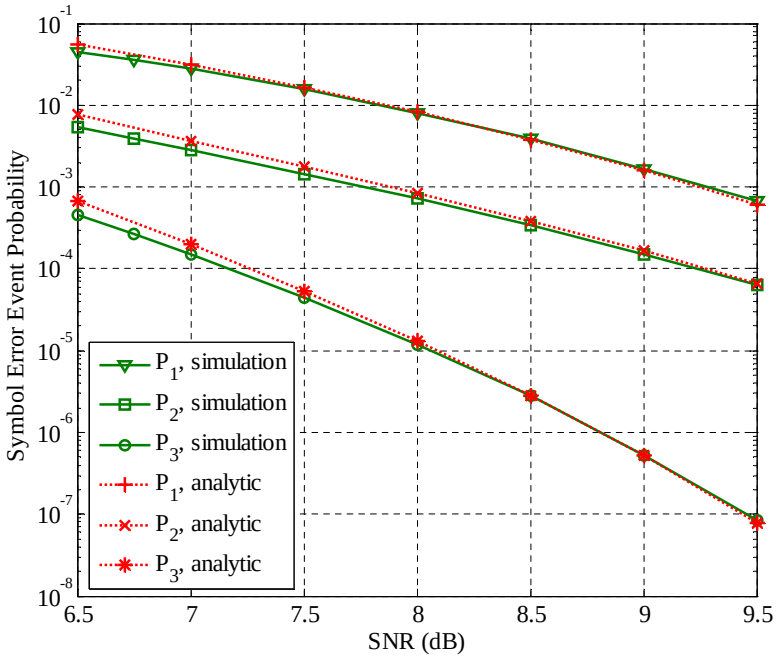


Figure 4.5: Comparison of simulation and analytic computation of symbol error event probabilities for $5 + 6D - D^3$ in AWGN, and 18-bit long symbols.

system. The analytic symbol error probabilities P_1, P_2, and P_3, calculated by (4.9) and (4.12), are in good agreement with simulation results. Figure 4.5 shows P_1, P_2, and P_3 for an 18-bit symbol based system, which corresponds to the tensor symbol length of TP-RS. The analytic and simulation results are again in good agreement. A comparison of the symbol error event probabilities at the input of single-level RS and component RS of TP-RS is shown in Figure 4.6. It is apparent that TP-RS sees a more sever channel compared to single level RS. The question remains, would the lower rate of two-level RS help combat the increased symbol error probability? The analytic estimation of SER helps answer this question for a wide range of code rates and block sizes. The analytic and simulation based symbol error event probabilities of Figure 4.4 are plugged in (4.16) to generate the analytic and semi-analytic SER, respectively. Shown in Figure 4.7, SER estimated by the analytic multinomial method agrees with the semi-analytic multinomial method for a wide range of correction powers t.

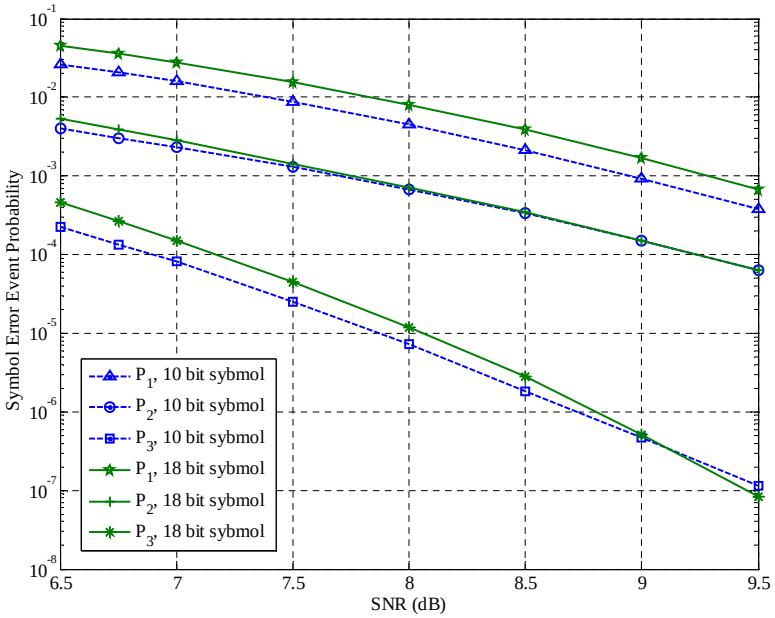


Figure 4.6: Simulation of symbol error event probabilities for $5 + 6D - D^3$ in AWGN, comparing and 18-bit and 10-bit symbol lengths.

Furthermore, Figure 4.8 shows that, compared to count-based simulation, the estimation error of the analytic method is uniform and small enough to allow the study of the relative performance gains of ideal TP-RS with respect to single-level RS. We study the performance advantage of TP-RS over single-level RS for a wide range of RS correction powers in Figure 4.9. There, we choose the minimum SNR required to achieve a SER of 10^{-13} as the performance benchmark and plot it as a function of RS correction power t. We also design an equal rate TP-RS at each t and plot its minimum SNR on the same graph. Then, we can see in this graph the effect of the severe rate penalty $1/R^2$ on codes with large t. To make it easier to compare the codes, we plot the difference in minimum SNR as function of rate in Figure 4.10. We can see in this figure that TP-RS furnishes more gain at higher code rates, which is the region of interest in recording applications. The recording industry is moving to the 4K sector size as the next generation standard. For that reason, it would be useful to examine wether TP-RS can still provide advantage at these longer sector sizes. To be able to support the 4K sector size we move to RS over $GF(2^{12})$, and

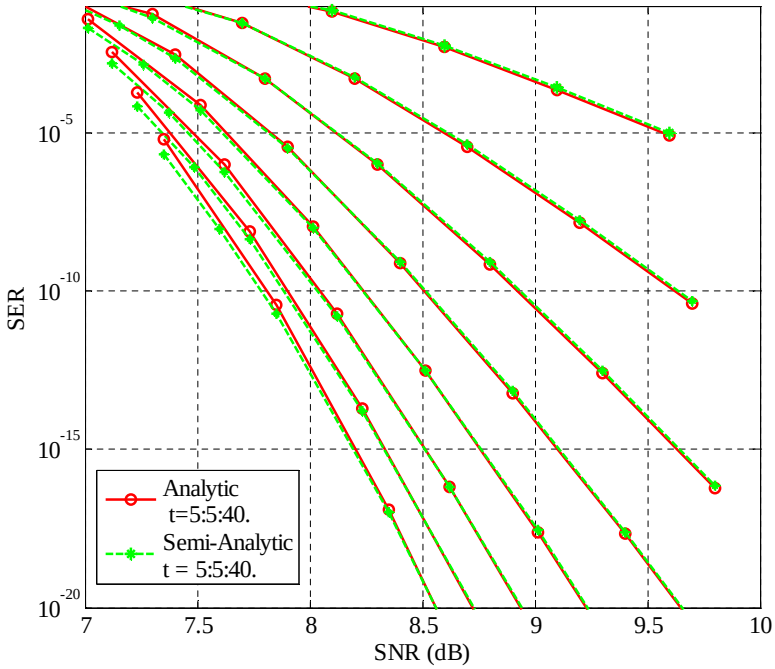


Figure 4.7: SER of $(450, 450-2t)$ RS , for $t = 5, 10, 15, 20, 25, 30, 35, 40$, and $5+6D-D^3$ in AWGN, comparing the semi-analytic and analytic methods.

compare it to the tensor product of $(24, 14)$ EPCC and RS over $GF(2^{10})$. In Figure 4.11, we find that 2K TP-RS can achieve the same or better minimum SNR compared to 4K RS. Moreover, ideal TP-RS achieves a uniform gain of 0.4 dB for all sector sizes.

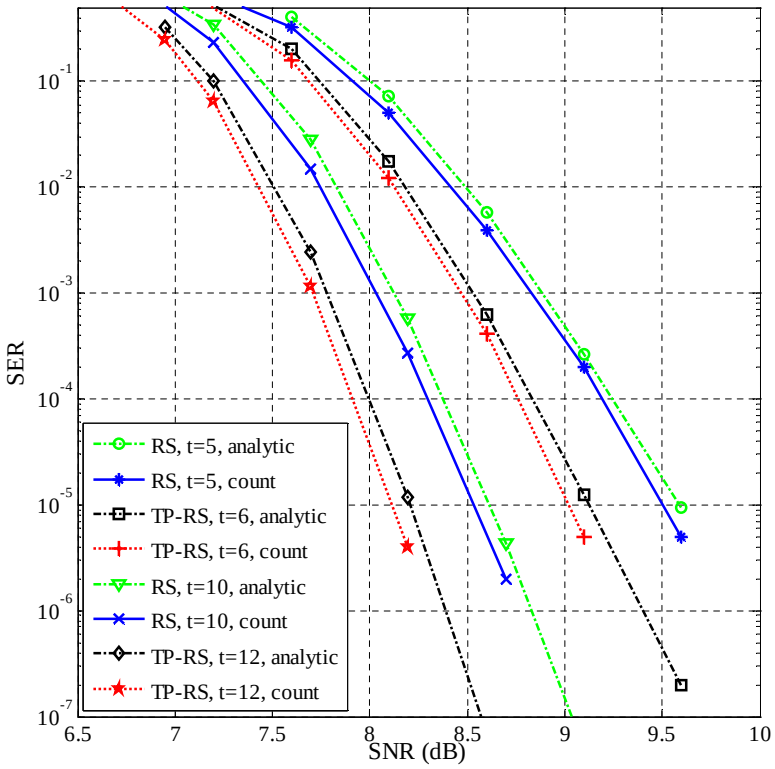


Figure 4.8: Analytic and count-based SER of shortened $(450, 450-2t)$ RS over $GF(2^{10})$, and an equal rate shortened $(250, 250-2t)$ TP-RS over $GF(2^8)$, for single-level RS correction powers $t = 5, 10$, and a $5 + 6D - D^3$ channel in AWGN.

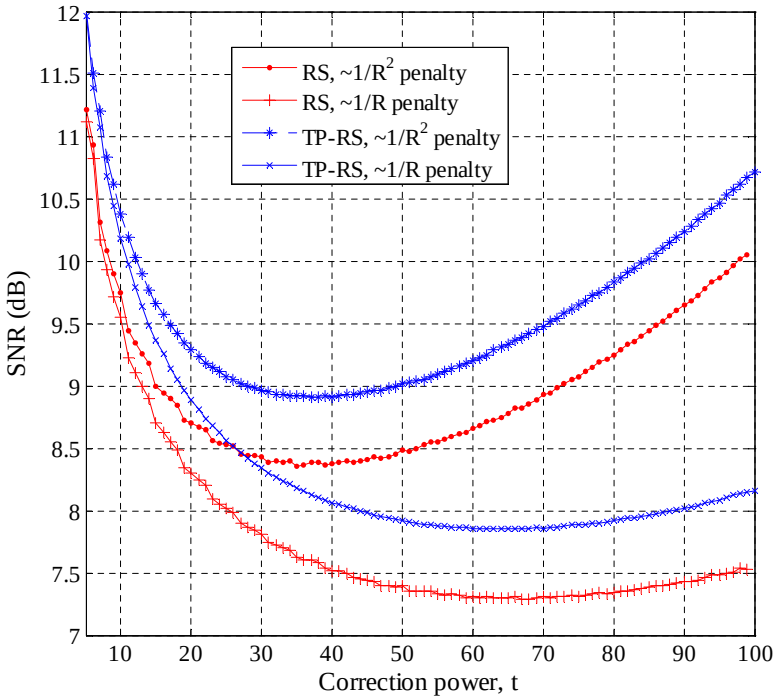


Figure 4.9: The analytic minimum SNR required to achieve a SER of 10^{-13} on ISI channel $5+6D-D^3$ corrupted by AWGN for the systems: $(450, 450-2t)$ RS over $GF(2^{10})$ from $t=5$ to $t=100$, compared to an equal rate shortened $(250, 250-2t)$ TP-RS over $GF(2^8)$. Results are shown for rate penalties $1/R$ dB and $1/R^2$ dB.

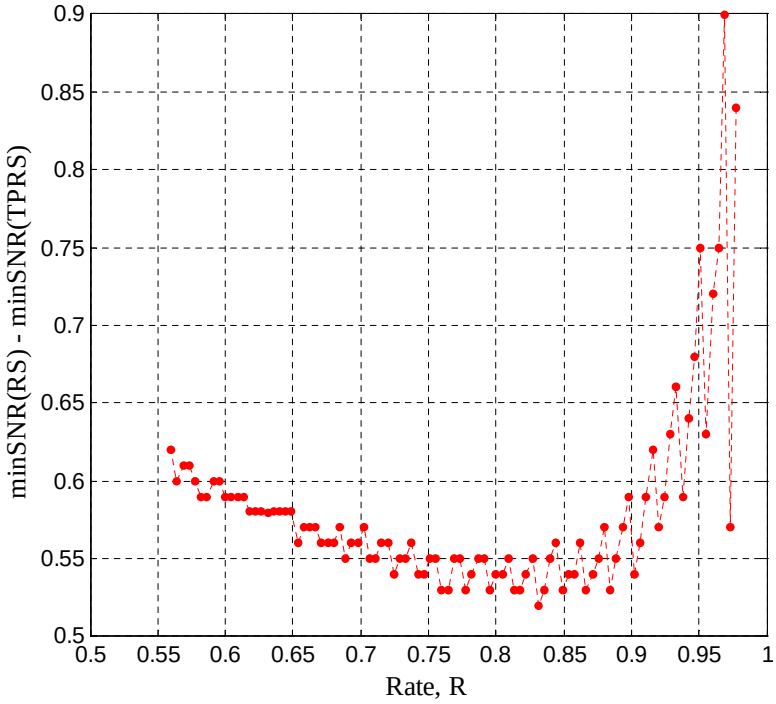


Figure 4.10: The difference between the analytic minimum SNR required to achieve a SER of 10^{-13} for RS and an equal rate TP-RS. Channel parameters are the same as in Figure 4.9.

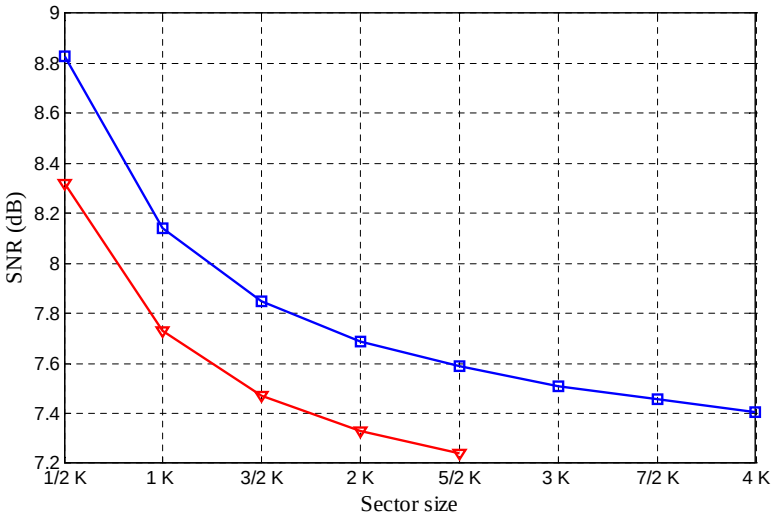


Figure 4.11: The analytic minimum SNR required to achieve a SER of 10^{-13} of RS with different sector sizes and rate 0.89 compared to an equal rate TP-RS. Channel parameters are the same as in Figure 4.9.

Chapter 5

Tensor LDPC Error-Pattern Correction Coding

Motivated by the need to design high rate codes that can bridge the gap to capacity for high density recording channels, we study replacing the industry standard Reed Solomon (RS) code with a channel-matched tensor-product concatenation of short EPCC and a symbol correcting LDPC code. The resulting tensor-product parity code possesses useful algebraic properties derived from its component codes. In particular, since EPCC is linear time encodable, and the component LDPC is also quasi-cyclic, TPPC is linear time encodable. Furthermore, if EPCC resolves error-pattern ambiguity per tensor symbol (i.e. EPCC codeword) through a large set of test patterns, TPPC iterative decoding will asymptotically approach the ML iterative decoder. The proposed iterative decoder relies on propagating the component soft-decoders' beliefs about the sequence of EPCC syndromes, which in turn represent a codeword for LDPC as a result of the construction method of TPPC.

In the operation of the TPPC iterative decoder, each potential syndrome value of a tensor symbol (EPCC codeword) has an accompanying list of probable error patterns and their likelihoods generated by the bank of error-pattern-matched correlators. The lists of error pattern likelihoods for all possible tensor symbol values map to a probability mass

function (p.m.f.) of the tensor-symbol syndromes, and the sequence of syndrome p.m.f's constitute channel-syndrome observations for the non-binary LDPC sum-product decoder. After the LDPC decoder converges to a zero syndrome codeword or reaches the maximum number of iterations, the sequence of enhanced syndrome probabilities are mapped into a sequence of most likely error patterns that are used to perturb the ML word for a second round of soft TPPC decoding. In addition, a global channel iteration is performed by generating bit-level reliabilities, using the set of candidate EPCC codewords, that are fed back as a priori information to the combination of channel detector and soft post-processor. The performance of this system in comparison to conventional coding schemes will be investigated.

5.1 T-EPCC-QLDPC Codes

We learned from the design of T-EPCC-RS that the component syndrome-correcting code's codeword is substantially reduced in length. Although the minimum distance is bound to be hurt if the increased redundancy does not compensate for the shorter code length, employing iterative soft decoding of the component syndrome-correcting code can recover performance if designed properly. While LDPC codes have strictly lower minimum distances compared to comparable rate and code length RS codes, the sparsity of its parity check matrix allows for effective belief propagation (BP) decoding. BP decoding of LDPC codes consistently performs better than the best known soft decoding algorithm for RS codes. Since the TPPC configuration results in 2 to 4 times reduction in component LDPC code length compared to a competing single level LDPC, a class of LDPC codes efficient at such short length are critical. LDPC codes on high order fields represent such good candidates. In that respect, [73] showed that the performance of binary LDPC codes in AWGN can be significantly enhanced by a move to fields of higher orders (extensions of $GF(2)$ being an example). Moreover, [73] established that for a monotonic improvement in waterfall

performance with field order, the parity check matrix for very short blocks has to be very sparse. Specifically, column weight 3 codes over $GF(q)$ exhibit worse bit-error-rate (BER) as q increases, whereas column weight 2 codes over $GF(q)$ exhibit monotonically lower BER as q increases. These results were later confirmed in [74], where they also showed through a density evolution study of large q codes that optimum degree sequences favor a regular graph of degree 2 in all symbol nodes. On the other hand, for satisfactory error floor performance, we found that using a column weight higher than 2 was necessary. This becomes more important as the minimum distance decreases for lower q. For instance, we found that a column weight of 3 improved the error floor behavior of $GF(64)$-LDPC at the expense of performance degradation in the waterfall region. Next, we introduce some definitions for the purpose of this chapter.

5.1.1 Notations and Definitions

- For a ceratin parity check matrix H corresponding to a linear code $\{C : Hc^t = 0 \forall c \in C\}$, a syndrome s is the non-zero range of a perturbation of a codeword $H(c+e)^t = s$. A "virtual syndrome" refers to the range of H for any bit block, not necessarily formed of data and parity bits.

- The multilevel log-likelihood ratio (mlLLR) of a random variable $\beta \in GF(q)$ corresponding to the probability mass function (p.m.f.) $p_i(\beta) = \Pr(\beta = i), \sum_{i=0}^{q-1} p_i(\beta) = 1$, can be defined as: $\gamma_i(\beta) = \log(\frac{p_i(\beta)}{p_0(\beta)}), \gamma_0(\beta) = 0$.

- $[\mathbf{x}]_i^j$ denotes a local segment $[x_i, x_{i+1}, ..., x_j]$ of the sequence x_k.

- Assume $\alpha_L = \log(\alpha)$ and $\beta_L = \log(\beta)$, then $(\alpha + \beta)_L = \log(e^{\alpha_L} + e^{\beta_L})$. Define $max^*(\alpha_L, \beta_L) = (\alpha + \beta)_L = \max(\alpha_L, \beta_L) + \log(1 + e^{-|\alpha - \beta|})$.

5.1.2 Design and Construction of QLDPC

The low rate and the relatively low column weight of TPPC's QLDPC makes the parity check matrix sparse enough, allowing the usage of high girth codes. To optimize the girth for a given rate, we employ the progressive edge growth (PEG) algorithm [74] in code design. PEG optimizes the placement of a new edge connecting a particular symbol node to a check node on the Tanner graph, such that the largest possible local girth is achieved. Furthermore, PEG construction is very flexible, allowing arbitrary code rates, Galois field sizes, and column weights. In addition, linear-time encoding can be achieved without noticeable performance degradation, resulting in linear time encodable tensor product codes. Of the two approaches to achieve linear time encodability, the upper triangular parity check matrix construction [74] and PEG Construction with a QC constraint [75], we choose the later approach because the designed codes have better error floor behavior. T-EPCC-LDPC lends itself to iterative soft decoding quite naturally. We present a low complexity soft decoder in the following sub section.

5.1.3 Example Designs of T-EPCC-QLDPC Codes

Example 1 Component codes:

- $(30, 22)$ EPCC over $GF(2)$, of rate 0.73.
- $(152, 95)$ LDPC over $GF(2^8)$, of rate 0.63.

TPPC is a binary $(4560, 4104)$ code, of $R = 0.9$, and 456 parity bits.

Example 2 Component codes:

- $(18, 10)$ EPCC over $GF(2)$, of rate 0.56.
- $(253, 196)$ LDPC over $GF(2^8)$, of rate 0.78.

TPPC is a binary $(4554, 4098)$ code, of $R = 0.9$, and 456 parity bits.

Example 3 Component codes:

- $(12, 6)$ EPCC over $GF(2)$, of rate 0.5.
- $(383, 303)$ LDPC over $GF(2^6)$, of rate 0.79.

TPPC is a binary $(4596, 4116)$ code, of $R = 0.9$, and 480 parity bits.

5.1.4 Soft Decoding of T-EPCC-QLDPC

A joint maximum likelihood (ML) decoder of the EPCC decoder with p parity bits and the matching ISI channel with memory L, has an exponential complexity in $p + L - 1$. We present a practical soft decoding scheme that separates the soft channel detection from EPCC decoding but, through the component LDPC in a TPPC setup, approaches the joint ML performance through channel iterations. The main stages of the decoder are (see Figure 5.2):

(1) **Detection postprocessing**:

- Utilizing *a priori* information from the previous decoding iteration, binary Viterbi generates the hard ML word based on channel observations, for which the error sequence is calculated and passed to the correlator bank.
- A bank of local correlators estimates the probability of dominant error type/location pairs for all positions inside each tensor symbol.

(2) **Syndrome p.m.f. calculation**:

- For each tensor symbol, the list of most likely error patterns are listed. This list includes single occurrences and a predetermined set of their combinations. The list is then divided into sublists, each under the virtual syndrome value it satisfies.

- For each tensor symbol, using each syndrome value's error likelihood list, we find the p.m.f. of the syndrome.

(3) **Q-ary LDPC decoding**:

- Using the sequence of syndrome probability mass functions, we decode the component Q-ary LDPC.
- For each tensor symbol, the LDPC-corrected syndrome p.m.f. is convolved with the syndrome p.m.f. at its input to generate the error syndrome p.m.f..

(4) **EPCC decoding**:

- For each tensor symbol, we find the list of most probable error syndromes and generate a list of test error words to satisfy each syndrome in the list.
- A bank of parallel EPCC single-error correcting decoders generate a list of most probable codewords along with their reliabilities.

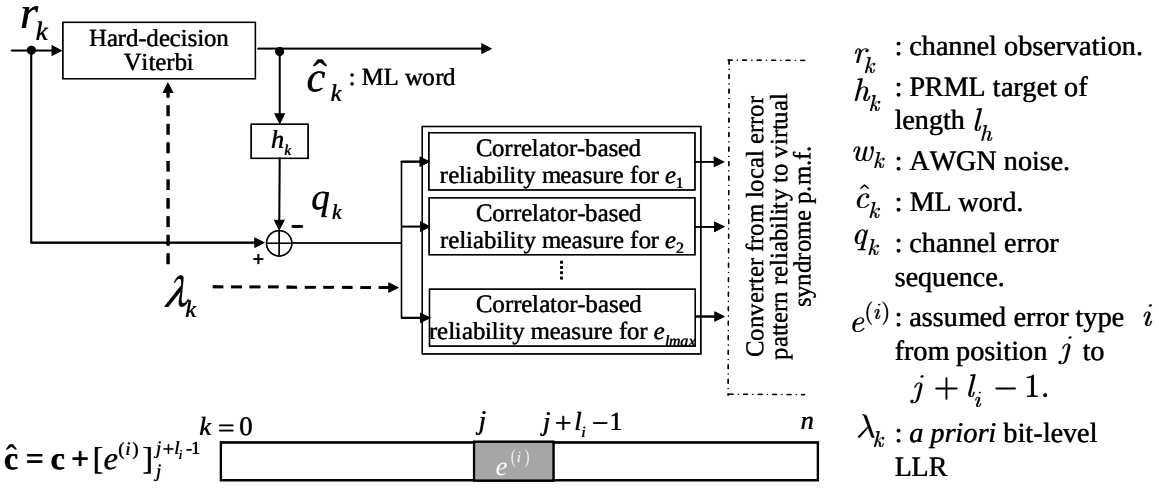
(5) **Bit-LLR feedback**:

- Using the codeword reliabilities we generate bit-level reliabilities that are fed back to the Viterbi detector and the detection postprocessing stage. Those bit-level reliabilities, serving as *a priori* information, favor paths which satisfy both the ISI and parity constraints.

We explain each of these steps in the following sub sections, but we replace any occurrence of syndrome p.m.f. by syndrome multi-level log-likelihood ratios (mlLLR), as decoding will be entirely in log domain for reasons explained below.

Detection Postprocessing

At this decoder stage we prepare a reliability matrix $\mathbf{C}(E)$ for error type/position pairs - captured in a tensor symbol of length l_T - that is usable by the next stage to calculate the



$$q_k = [\mathbf{e}^{(i)}]_j^{j+l_i-1} * h_k + w_k = [\mathbf{s}^{(i)}]_j^{j+l_i^h} + w_k$$

Figure 5.1: Bank of parallel error-matched correlators to find error pattern type/position reliabilities.

tensor symbol's syndrome mlLLR:

$$\mathbf{C}(E) = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ l_{max} \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & \cdots & l_T \end{array} \\ \begin{bmatrix} C(e_0^{(1)}) & C(e_1^{(1)}) & \cdots & C(e_{l_T-1}^{(1)}) \\ C(e_0^{(2)}) & C(e_1^{(2)}) & \cdots & C(e_{l_T-1}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ C(e_0^{(l_{max})}) & C(e_1^{(l_{max})}) & \cdots & C(e_{l_T-1}^{(l_{max})}) \end{bmatrix} \end{array} \tag{5.1}$$

where $C(e_k^{(i)})$ is the error pattern (type i / position k) reliability measure computed by the maximum *a posteriori* (MAP)-based error-pattern correlator shown in Figure 5.1. The bank of local correlators discussed here was also employed in [40] for AWGN channels, and in [39] for data-dependent noise environments. We now discuss how to generate these local metrics. Let r_k be the channel detector input sequence $r_k = c_k * h_k + w_k$, where c_k is the bipolar representation of the recorded codeword sequence, h_k is the partial response channel of length l_h, and w_k is zero-mean AWGN noise with variance σ^2. Also, let $q_k = r_k - (\hat{c}_k * h_k) = (c_k - \hat{c}_k) * h_k + w_k$ be the channel detector's output error sequence. If a

target error pattern sequence $e_k^{(i)}$ occurs at positions from $k = j$ to $k = j + l_i - 1$, then q_k can be written as

$$\begin{aligned} q_k &= (c_k - \hat{c}_k) * h_k + w_k \\ &= [\mathbf{e}^{(i)}]_j^{j+l_i-1} * h_k + w_k \\ &= [\mathbf{s}^{(i)}]_j^{j+l_i^h} + w_k. \end{aligned} \tag{5.2}$$

Where $s_k^{(i)}$ is the channel response of the error sequence given by $s_k^{(i)} = e_k^{(i)} * h_k$, and $l_i^h = l_i + l_h - 2$. Note that we define the start of the tensor symbol at $j = 0$. So, if $j < 0$, then, the error pattern starting position is in a preceding tensor symbol.

The reliability for each error pattern with starting position, j, can be computed by the local *a posteriori* probabilities (ignoring tensor symbol boundaries for now):

$$\Pr\left([s^{(i)}]_j^{j+l_i^h} \middle| [r]_j^{j+l_i^h}, [\hat{\mathbf{c}}]_{j-l_h+1}^{j+l_i^h}\right) = \Pr\left([s^{(i)}]_j^{j+l_i^h} \middle| [q]_j^{j+l_i^h}, [\hat{\mathbf{c}}]_{j-l_h+1}^{j+l_i^h}\right). \tag{5.3}$$

The most likely error type/position pair in a tensor symbol maximizes the a posteriori probability ratio of its reliability to the reliability of the second most probable error event (the competing event in this case would be the ML word itself). The ratio to maximize is

$$\frac{\Pr\left(e_j^{(i)} \middle| \hat{\mathbf{c}}\right)}{\Pr\left([\text{ML word}]_j^{j+l_i^h} \middle| \hat{\mathbf{c}}\right)} = \frac{\Pr\left([\mathbf{q}]_j^{j+l_i^h} \middle| [\hat{\mathbf{c}}]_{j-l_h+1}^{j+l_i^h}, [\mathbf{s}^{(i)}]_j^{j+l_i^h}\right) \Pr\left([\mathbf{s}^{(i)}]_j^{j+l_i^h}\right)}{\Pr\left([\mathbf{q}]_j^{j+l_i^h} \middle| [\hat{\mathbf{c}}]_{j-l_h+1}^{j+l_i^h}, [\tilde{\mathbf{s}}^{(i)}]_j^{j+l_i^h}\right) \Pr\left([\tilde{\mathbf{s}}^{(i)}]_j^{j+l_i^h}\right)}, \tag{5.4}$$

where $[\tilde{\mathbf{s}}^{(i)}]_j^{j+l_i^h}$ is the ML word's clean channel response. Given the noise model, $[\mathbf{q}]_j^{j+l_i^h}$ is a sequence of independent Gaussian random variables with variance σ^2. Therefor, maximizing (5.4) can be shown to be equivalent to maximizing the log-likelihood local measure [40]:

$$C(e_j^{(i)}) = \sum_{k=j}^{j+l_i^h} \frac{1}{2\sigma^2}\left(q_k^2 - (q_k - s_k^{(i)})^2\right) - \log \frac{\Pr([\tilde{s}^{(i)}]_j^{j+l_i^h})}{\Pr([s^{(i)}]_j^{j+l_i^h})} \tag{5.5}$$

where the *a priori* bias in (5.5) is evaluated as:

$$\log \frac{P([\tilde{s}^{(i)}]_j^{j+l_i^h})}{P([s^{(i)}]_j^{j+l_i^h})} = \sum_{k=j,\hat{c}_k=+1}^{j+l_i^h} \lambda_k - \sum_{k=j,\hat{c}_k=-1}^{j+l_i^h} \lambda_k. \tag{5.6}$$

Equation (5.5) represents the "local" error-pattern correlator output in the sense that it essentially describes the correlator operation between q_k and the channel output version of the dominant error pattern $e_j^{(i)}$ within the local region $[j, j+l_i^h]$. However, equation (5.5) ignores that errors can span tensor symbol boundaries when $j < 0$ or $j > l_T - 1$. Hence, to calculate a modified metric relevant to the current tensor symbol, we utilize the following procedure:

- $\forall i$ at $j = 0$, modify $C(e_0^{(i)})$

$$\tilde{C}(e_0^{(i)}) = \overset{l_{max}}{\underset{k=i}{max^*}} \left(C(e_{-k+i}^{(k)}) \right),$$

  independently for each i, where l_{max} is the maximum length error pattern.

- Starting at $i = 1$ and $j = l_T - 1$, do:

  (i) $\tilde{C}(e_j^{(i)}) = \overset{l_{max}}{\underset{k=i}{max^*}} \left(C(e_j^{(k)}) \right)$,

  (ii) $\forall k > i$, set $C(e_j^{(k)}) = -\infty$.

  (iii) Set $i = i + 1$, $j = j - 1$.

  (iv) If $i < l_{max}$ go back to (i).

We assume here that dominant error events span only two tensor symbols at a time and that they do not include error free gaps, which is certainly true for the case study of this work.

Syndrome mlLLR Calculation

For each tensor symbol i, utilizing $\tilde{\mathbf{C}}(E)$, we need to find the p.m.f. or the log domain mlLLR of its virtual syndrome $\mathrm{Syn_i} \in GF(2^{p_{epcc}})$, for EPCC with p_{epcc} parity bits. To limit the computational complexity of this calculation, we construct a syndrome only from the dominant errors and a subset of their multiple occurrences. Denote $\hat{Pr}(Syn_i = \alpha^{j-1})$ as the running estimate of the p.m.f.'s j-th bin, and $\hat{\gamma}(Syn_i = \alpha^{j-1}) = \log(\hat{Pr}(Syn_i = \alpha^{j-1}))$ as the running estimate of mlLLR. Denote a one dimensional index of $\tilde{\mathbf{C}}(E)$ as

$p^{rc} = (p_c \times l_{max}) + p_r$ corresponding to the p_r-*th* row and p_c-*th* column of $\tilde{\mathbf{C}}(E)$ and error $E(p^{rc})$. We choose the dominant list as the L patterns with the largest corresponding elements of $\tilde{\mathbf{C}}(E)$ having indexes $\{p_i^{rc}\}_{i=1}^{i=L}$. Based on this list, we developed the following procedure to compute $\hat{\gamma}(Syn_i = j)$:

- Step 1 (Single occurrences):

$$\begin{aligned} \hat{\gamma}(Syn_i = \alpha^{j-1}) \quad &= \quad \underset{k=p_1^{rc}}{\overset{p_L^{rc}}{max^*}} \left(\tilde{C}[k]\right), \\ &\forall k: \quad G_f(H_{epcc} \times [\hat{c}_{i \times l_T}^{(i+1)\times l_T - 1} \oplus E(k)]^t) = \alpha^{j-1}. \end{aligned}$$

- Step 2 (Double occurrences):

$$\begin{aligned} \hat{\gamma}(Syn_i = \alpha^{j-1}) = max^*(\hat{\gamma}(Syn_i = \alpha^{j-1}), \underset{k=p_1^{rc}, m=p_1^{rc}}{\overset{p_L^{rc}, p_L^{rc}}{max^*}} \left(\tilde{C}[k] + \tilde{C}[m]\right)), \\ \forall \{k, m\} : G_f(H_{epcc} \times [\hat{c}_{i \times l_T}^{(i+1)\times l_T - 1} \oplus E(k)]^t) \\ \underset{GF(q_{epcc})}{\oplus} G_f(H_{epcc} \times [\hat{c}_{i \times l_T}^{(i+1)\times l_T - 1} \oplus E(m)]^t) = \alpha^{j-1}, \\ \underset{k \neq m}{\mathbb{D}}(E(k), E(m)) > E_{free}. \end{aligned}$$

  where $q_{epcc} = 2^{p_{epcc}}$, $\mathbb{D}$ is the error free distance between two errors, $E_{free} = l_h - 1$ is the error free distance of the channel beyond which errors are independent and $G_f(...)$ is an operator that maps p_{epcc}-bit vectors into $GF(2^{p_{epcc}})$ symbols.

- ...

- Step M (M occurrences):

$$\begin{aligned}\hat{\gamma}(Syn_i = \alpha^{j-1}) \;=\; & max^*(\hat{\gamma}(Syn_i = \alpha^{j-1}), \\ & \underset{q_1=p_1^{rc},q_2=p_1^{rc},...,q_M=p_1^{rc}}{\overset{p_L^{rc},p_L^{rc},...,p_L^{rc}}{max^*}} \Big(\tilde{C}[q_1] + \tilde{C}[q_2] + ... + \tilde{C}[q_M]\Big)) \\ & \forall\{q_1, q_2, ..., q_M\} : \\ & \bigoplus_{m=1,GF(q_{epcc})}^{m=M} G_f(H_{epcc} \times [\hat{c}_{i\times l_T}^{i+1\times l_T} \oplus E(q_m)]^t) = \alpha^{j-1}, \\ & \underset{s,t,s\neq t}{\mathbb{D}}(E(q_s), E(q_t)) > E_{free}.\end{aligned}$$

- Step $M+1$ (0-*th* syndrome; so that the corresponding p.m.f. sums to 1):

$$\hat{\gamma}(Syn_i = \alpha^{-\infty}) = max^*(\hat{\gamma}(Syn_i = \alpha^{-\infty}), -max^*(0, \overset{q_{epcc}-1}{\underset{j=1}{max^*}}(\hat{\gamma}(Syn_i = \alpha^{j-1})))) \tag{5.7}$$

- Step $M+2$ (Normalization):

$$\gamma(Syn_i = \alpha^{j-1}) = \hat{\gamma}(Syn_i = \alpha^{j-1}) - \hat{\gamma}(Syn_i = \alpha^{-\infty}). \tag{5.8}$$

In steps 1 through M, to calculate the log-likelihood of syndrome i assuming value α^{j-1}, we sum the probabilities of all single and multiple errors in the ML word whose virtual syndromes equal α^{j-1}. This is equivalent to performing the max^* operation in log domain on error reliabilities dictated by $\tilde{\mathbf{C}}\,(E)$. However, to limit the complexity of this stage, we use a truncated set of error combinations, in all steps from 1 to M. Also, for syndrome values that do not correspond to any of the combinations, we set their reliability to $-\infty$. Since there are many such syndrome values, the corresponding constructed p.m.f. would be sparse.

Q-ary LDPC Decoding

Now, the sequence of syndrome mlLLRs is passed as multi-level channel observations to the QLDPC decoder. We choose to implement the log-domain Q-ary fast Fourier transform-

belief propagation (FFT-BP) decoder in [76] for this purpose. The choice of log-domain decoding is essential, since if we use the syndrome p.m.f. as input, BP would run into numerical instability resulting from the sparse p.m.f. generated by the preceding stage.

The LDPC output *posteriori* mlLLRs correspond to the virtual syndromes of tensor symbols, rather than the syndromes of errors needed by EPCC. Similar to the decoder of T-EPCC-RS, error syndromes Syn_i^e are the finite field sum of the LDPC's channel input, Syn_i^{ch}, and output *posteriori* syndromes, Syn_i^p. Moreover, the addition of hard syndromes corresponds to the convolution of their p.m.f.s, and this convolution in probability domain corresponds to the following operation in log-domain:

$$\begin{aligned}\hat{\gamma}(Syn_i^e = \alpha^{\beta_e}) \;\; = \;\; & max^*(\hat{\gamma}(Syn_i^e = \alpha^{\beta_e}), \gamma(Syn_i^{ch} = \alpha^{\beta_{ch}}) + \gamma(Syn_i^p = \alpha^{\beta_p})) \\ & \forall \beta_e : \alpha^{\beta_e} = \alpha^{\beta_{ch}} \underset{GF(q_{epcc})}{\oplus} \alpha^{\beta_p}, \\ & \beta_{ch} = -\infty, 0, ..., q_{epcc} - 2; \beta_p = -\infty, 0, ..., q_{epcc} - 2\end{aligned}$$

which is later normalized, similar to LDPC BP mlLLR message normalization, according to:

$$\gamma(Syn_i^e = \alpha^{\beta_e}) = \hat{\gamma}(Syn_i^e = \alpha^{\beta_e}) - \hat{\gamma}(Syn_i^e = \alpha^{-\infty}), \forall \beta_e = -\infty, 0, ..., q_{epcc} - 2. \quad (5.9)$$

EPCC Decoding

An error syndrome will decode to many possible error events due to the low minimum distance of single-error correcting EPCC. However, EPCC relies on local channel side information to implement a list-decoding-like procedure that enhances its multiple error correction capability. Moreover, the symbol-level code length reduces the probability of such multiple error occurrences considerably. To save power consumption, EPCC is turned on for a tensor symbol i only if the most likely value in the syndrome mlLLR is nonzero, i.e., $\underset{\alpha^\beta \in GF(q_{epcc})}{\arg\max} \; \gamma(Syn_i^e = \alpha^\beta) \neq \alpha^{-\infty}$, indicating that a resolvable error has occurred.

After this, a few syndrome values, 3 in our case, most likely according to the mlLLR, are decoded in parallel. For each of these syndromes, the list decoding algorithm goes as [40][39]:

- A test error word list is generated by inserting the most probable combination of local error patterns into the ML tensor symbol.
- An array of parallel EPCC single-pattern correcting decoders decodes the test words to produce a list of valid codewords that satisfy the current syndrome.
- The probability of a candidate codeword is computed as the sum of likelihoods of its parent test-word and the error pattern separating the two.
- Each candidate codeword probability is biased by the likelihood of the error syndrome it is supposed to satisfy.

In the generation of test words, we only combine independent error patterns that are separated by the error free distance of the ISI channel.

Soft Bit-level Feedback Calculation

The list of candidate codewords and probabilities are used to generate bit level-probabilities in a similar manner to [49] [39]. The conversion of word-level reliability into bit-level reliability for a given bit position can be done by grouping the candidate codewords into two groups, according to the binary value of the hard decision bit in that bit position, and then performing group-wise summing of the word-level probabilities. Three scenarios are possible for this calculation:

(i) The candidate codewords do not all agree on the bit decision for location k; then, given the list of codewords and their accompanying *a posteriori* probabilities, the reliability λ_k of the coded bit c_k is evaluated as

$$\lambda_k = \log \frac{\sum_{\mathbf{c} \in \mathbf{S}_k^+} Pr(\mathbf{c}/\hat{\mathbf{c}}, \mathbf{r})}{\sum_{\mathbf{c} \in \mathbf{S}_k^-} Pr(\mathbf{c}/\hat{\mathbf{c}}, \mathbf{r})} \tag{5.10}$$

where $\mathbf{S}_k^+$ is the set of candidate codewords where $c_k = +1$, and $\mathbf{S}_k^-$ is the set of candidate codewords where $c_k = -1$

(ii) Although rare for such short codeword lengths, in the event that all codewords do agree on the decision for c_k, a method inspired by [49] is adopted for generating soft information as follows

$$\lambda_k = \beta^{iter} \times \lambda_{max} \times \hat{d}_k \tag{5.11}$$

where $\hat{d}_k$ is the bipolar representation of the agreed-upon decision, λ_{max} is a pre-set value for the maximum reliability at convergence of turbo performance, and the multiplier $\beta^{iter} < 1$ is a scaling factor. $\beta^{iter} \ll 1$ in the first global iterations and is increased to 1 as more global iterations are performed and the confidence in bit decisions improved. Thus, this back-off control process reduces the risk of error propagation.

(iii) The heuristic scaling in (5.11) is again useful when EPCC is turned off for a tensor symbol, in case of the most likely syndrome being 0. Then, the base hard value of the tensor symbol corresponds to the most likely error event found as a side product in step 3 of the T-EPCC-QLDPC decoder.

Stopping Criterion for T-EPCC-QLDPC and RS Erasure Decoding

Due to the ambiguity in mapping syndromes to errors in the first stage, the possibility of non-targeted error patterns, or errors that have zero syndromes that are transparent to H_{epcc}, a second line of defense is essential to take care of undetected errors. Therefore, an outer RS code of small correction power t_{out} is concatenated to T-EPCC-QLDPC to take care of the imperfection of the component EPCC. Several functions are offered by this code, including:

- *Stoping Flag*: If the RS syndrome is zero, then, global iterations are halted and decisions are released.

- *Outer ECC*: Attempt to correct residual errors at the output of EPCC after each global iteration.

- *Erasure Decoding*: If the RS syndrome is nonzero, then, for those tensor symbols for which EPCC was turned on, declare their bits as erasures. Next, find the corresponding RS symbol erasures, and attempt RS erasure decoding which could correct up to $2 \times t_{out}$ such erasures. In this case, T-EPCC acts as an error locating code.

5.2 Simulation Results and Discussion

We compare three coding systems based on LDPC: conventional binary LDPC, Q-ary LDPC, and T-EPCC-QLDPC. We study their sector error rate (SER) performance on an ISI channel with response $1 + 0.85D$ corrupted by AWGN, and with coding rate penalty $10 \log_{10}(1/R)$. The nominal systems run at a coding rate of 0.9. The minimum SNR required to achieve reliable recording at this rate is 3.9 dB, estimated as in [77].

5.2.1 Single-level BLDPC & QLDPC Simulation Results

In Figure 5.3, we compare SER of the following LDPC codes, each constructed by PEG with a QC constraint:

- A $(4550, 4095)$ $GF(2)$-LDPC, of column weight 5, and cycle size 91. The channel detector is a 2 state binary BCJR.

- A $(570, 510)$ $GF(2^8)$-LDPC, of codeword length 4560 bits, column weight 2, and cycle size of 15 symbols. The channel detector is a symbol-BCJR with 256 branches emanating from each of 2 states.

- A $(760, 684)$ $GF(2^6)$-LDPC, of codeword length 4560 bits, column weight 2, and cycle size of 19 symbols. The channel detector is a symbol-BCJR with 64 branches emanating from each of 2 states.

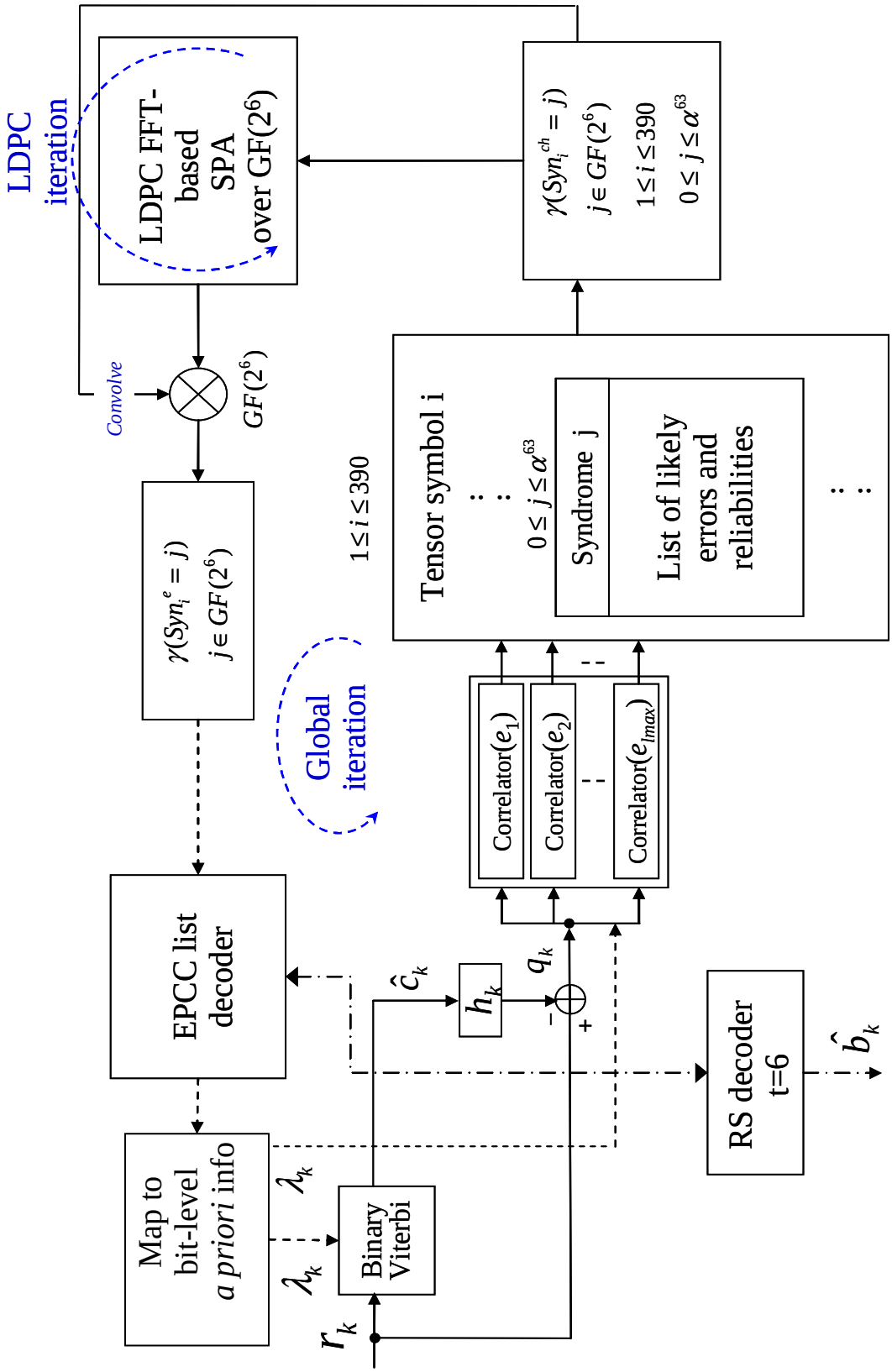


Figure 5.2: T-EPCC-QLDPC soft decoder of $(12, 6, 6)$ EPCC $\otimes$ $(390, 312)$ $GF(2^6)$-LDPC.

- A $(775, 700)$ $GF(2^6)$-LDPC, of codeword length 4650 bits, column weight 3, and cycle size of 25 symbols. The channel detector is a symbol-BCJR with 64 branches emanating from each of 2 states.

For the binary LDPC turbo equalizer, we run a maximum of 10×50 iterations, 10 global, and 50 LDPC BP iterations. For the Q-ary turbo equalizers, on the other hand, we run a maximum of 0×50 iterations. A column weight of 2 gives the best waterfall performance of Q-ary LDPC. However, $GF(2^6)$-LDPC exhibits an error floor as early as at 6×10^{-4}, whereas a higher order field of $GF(2^8)$ does not show such a tendency down to 5×10^{-5}. Nevertheless, the prohibitive complexity of $GF(2^8)$ symbol-BCJR makes $GF(2^6)$ LDPC a more attractive choice. Still, we need to sacrifice $GF(2^6)$-LDPC's waterfall performance gains to guarantee a lower error floor. For that purpose, we move to a column weight 3 $GF(2^6)$-LDPC that is 1.37 dB away at 1×10^{-4} from the independent uniformly-distributed capacity $C_{I.U.D.}$ of the channel [77], and 0.4 dB away from $GF(2^8)$ LDPC. We have observed that while binary LDPC can gain up to 0.4 dB through 10 channel iterations before gain saturates, $GF(2^8)$-LDPC and $GF(2^6)$-LDPC achieve very little iterative gain by going back to the channel. One way to explain this phenomenon, is that symbol-level LDPC decoding divides the bit stream into LDPC symbols that capture the error events introduced by the channel detector, rendering the binary intersymbol interference limited channel into a memoryless multi-level AWGN limited channel. Nonetheless, error events spanning symbol boundaries reintroduce correlations between LDPC symbols that are broken only by going back to the channel. In other words, if it was not due to such boundary effects, a Q-ary LDPC equalizer would not exhibit any iterative turbo gain whatsoever. Nonetheless, bi-directional symbol BCJR is way too complex to justify salvaging the small iterative gain by performing extra channel iterations [78]. This is where error event matched decoding comes into the picture, which leads us to the results of the next section.

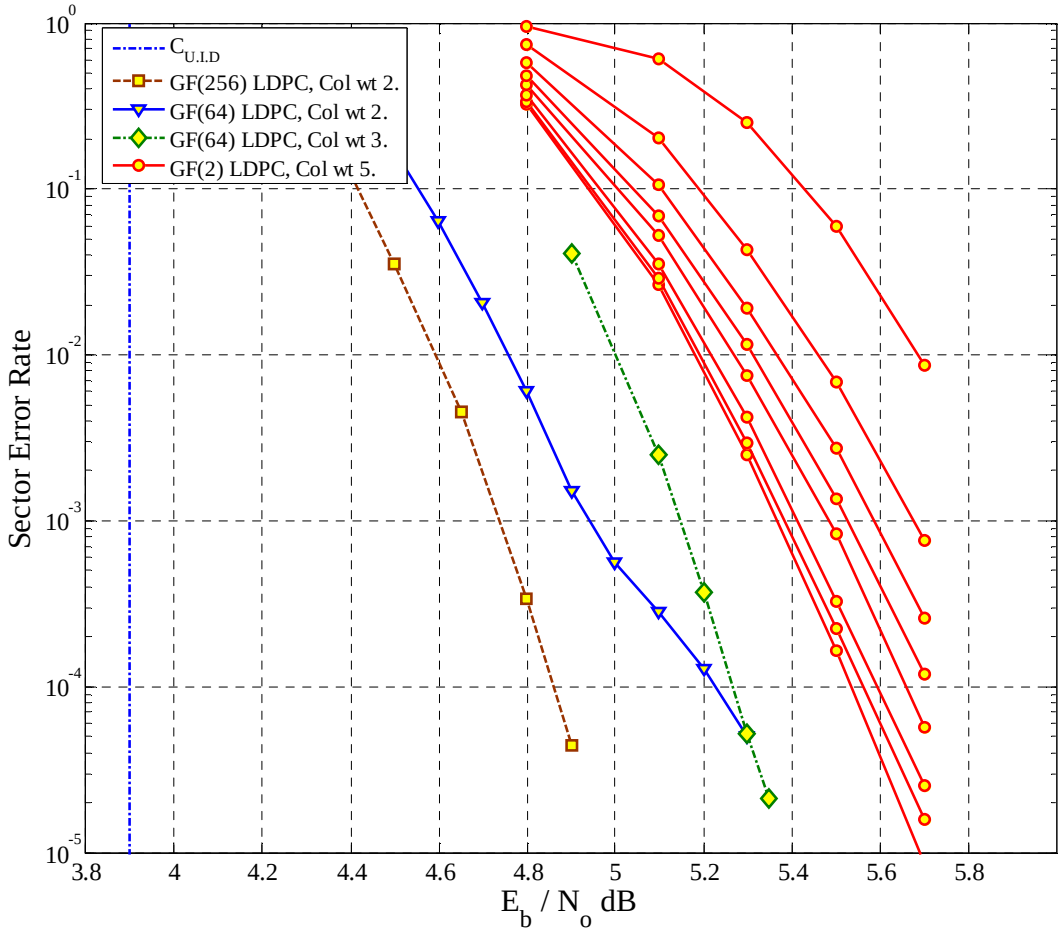


Figure 5.3: Comparing SER of: $\{0 \sim 10\} \times 50$ iterations of binary LDPC, 0×50 iterations of $GF(2^8)$ LDPC of column weight 2, and 0×50 iterations of $GF(2^6)$ LDPC of column weights 2 and 3. Minimum SNR to achieve a reliable coding rate of 0.9 is 3.9 dB for $1 + 0.85D$.

5.2.2 T-EPCC-QLDPC Simulation Results

We construct two T-EPCC-QLDPC codes of rate 0.9. Based on a component EPCC $(12, 6)$ of rate 0.5, they are:

- $1/2$KB T-EPCC-QLDPC: A binary $(4680, 4212)$ TPPC, of rate 0.9, and 468 parity bits, based on a component $(390, 312)$ PEG-optimized QC $GF(2^6)$ LDPC, of rate 0.8, column weight 3, and cycle size 26.

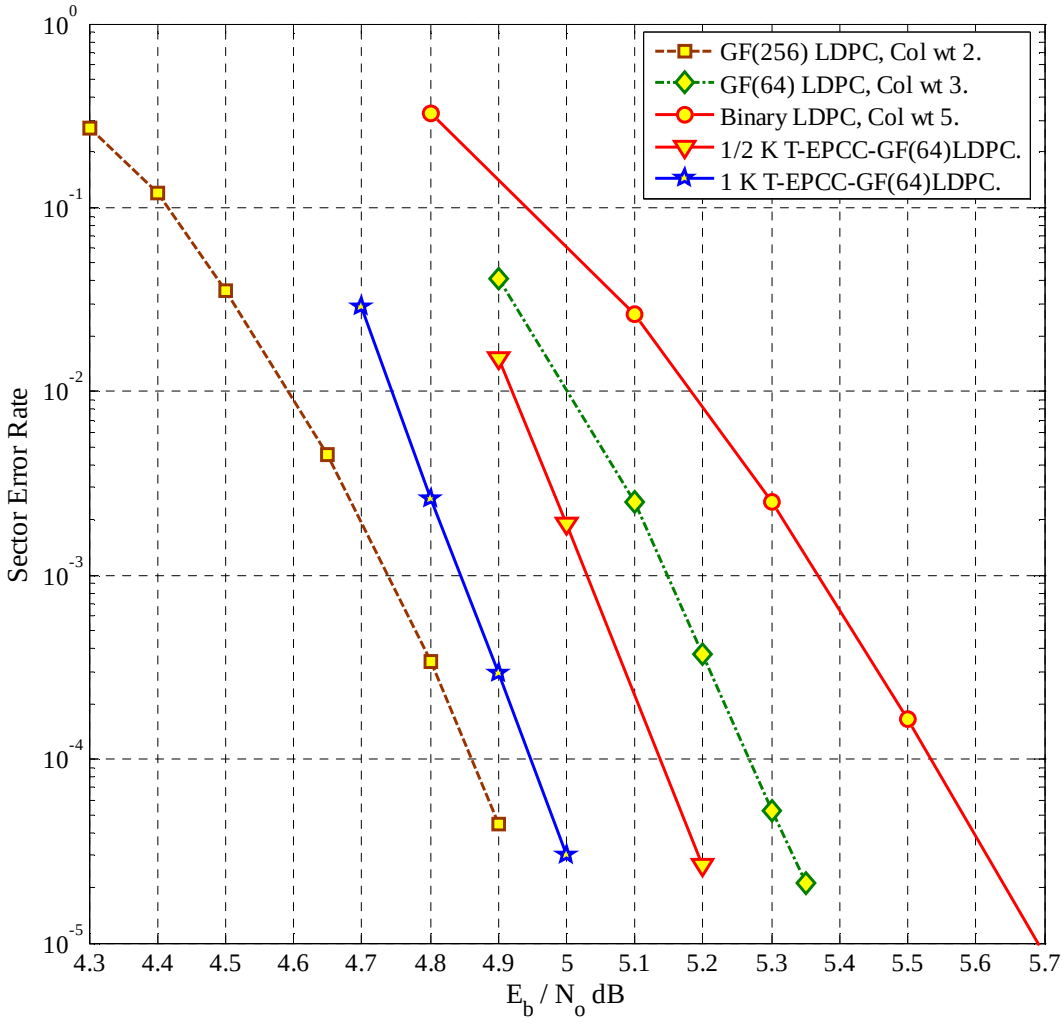


Figure 5.4: Comparing SER of: 10×50 iterations of binary LDPC, 0×50 iterations of $GF(2^8)$ LDPC of column weight 2, 0×50 iterations of $GF(2^6)$ LDPC of column weight 3, and 0×50 iterations of ideal $1/2$KB and 1KB T-EPCC-$GF(2^6)$LDPC based on column weight 3 LDPC.

- 1KB T-EPCC-QLDPC: A binary $(9360, 8424)$ TPPC, of rate 0.9, and 936 parity bits,based on a component $(780, 624)$ PEG-optimized QC $GF(2^6)$ LDPC, of rate 0.8, column weight 3, and cycle size 52.

The control mechanism of iterative decoding for these codes is as follows: if EPCC results in less than 6 RS symbol errors for $1/2K$ or less than 12 for 1KB, or if EPCC generates more errors than this, but declares less than 12 erasures for $1/2K$ or 24 erasures for 1KB, then, decoding halts and decisions are released. Otherwise, one more channel iteration is done by passing EPCC soft bit-level LLR's to Viterbi detection and the bank of error-matched correlators.

First, we study the SER of T-EPCC-QLDPC up to the component $GF(2^6)$ LDPC, at the first channel pass, which is function of the Viterbi ML word, accuracy of generating syndrome mlLLR, and actual LDPC code strength. This SER represents the best that the TPPC code can do, under the assumption of perfect component EPCC, i.e., as long as QLDPC generates a clean syndrome-codeword, then EPCC generates a clean data-codeword. Figure 5.4 shows the ideal SER of the two TPPC codes, assuming perfect EPCC, compared to single-level $GF(2^6)$ and $GF(2^8)$ LDPC. Ideal $1/2$KB TPPC is 0.15 dB better than single level $GF(2^6)$ LDPC at 3×10^{-5} SER. In $1/2$KB TPPC, the component $GF(2^6)$ LDPC has half the codeword length of the single level counterpart, saving 50% of the decoder complexity, while still furnishing gain. The component QLDPC faces a harsher channel than single-level QLDPC, because the symbol error probability of 6-bit data symbols is strictly less than 6-bit syndrome-symbol error probability, and syndrome symbols are compressed down from 12-bit data symbols. Also, the decrease in codeword length of component QLDPC affects its minimum distance. Still, these impairments are effectively compensated for by an 11.4% increase in redundancy for the component LDPC. If we match the codeword lengths of TPPC's component LDPC to single-level LDPC, as part of constructing 1KB TPPC, then, 1KB TPPC will have similar decoder complexity to $1/2$KB single-level LDPC with a 0.35 dB SNR advantage for TPPC at 3×10^{-5} SER.

Due to the imperfections of EPCC design, including miscorrection due to one-to-many syndrome to error position mapping, and undetected errors due to EPCC's small minimum distance, achieving the ideal performance in Figure 5.4 is not possible in one channel pass.

In the implementation of the full T-EPCC-$GF(2^6)$LDPC decoder, we run up to 3 channel iterations, and include a $t = 6$ outer RS for the $1/2$KB case, and a $t = 12$ outer RS for the 1KB case, so as to protect against EPCC residual errors. Simulation results in Figure 5.5 for a noise environment of rate penalty $10\log_{10} 1/R$, demonstrate that after 3 channel iterations, the ideal and practical performances lock, while incurring an SNR degradation of about 0.1 dB resulting from the rate penalty of outer RS. Also, $1/2$KB TPPC saves 50% of decoder complexity while achieving the same performance as single level LDPC, whereas 1KB TPPC has the same decoding complexity as single-level LDPC while furnishing 0.2 dB gain at 3×10^{-5} SER. In Figure 5.6, we properly shift the TPPC codes to the right to reflect performance in the more severe noise environment of rate penalty $10\log_{10} 1/R^2$. Now, $1/2$ KB T-EPCC-$GF(2^6)$LDPC is 0.12dB better than $GF(2)$LDPC and 0.13 worse than $GF(64)$-LDPC at 3×10^{-5} SER, and thus represents a tradeoff between the lower complexity of $GF(2)$LDPC and performance advantage of $GF(2^6)$-LDPC. On the other hand, the 1KB TPPC still provides a gain of 0.07dB with respect to $GF(64)$-LDPC at the same SER.

In terms of implementation complexity, the complexity and latency of bi-directional $GF(2^6)$ BCJR in the single level code far exceeds the overall complexity of the non-LDPC parts of two level T-EPCC-$GF(2^6)$LDPC. We also argue that performing 3 global iterations of T-EPCC-$GF(2^6)$LDPC can be done within one sector time by borrowing time from subsequent sectors, and as such, we do not require hardware duplication to achieve back to back decoding. Hence, the complexity comparison with $GF(2^6)$-LDPC performing 0 global iterations still holds when performing 3 global iteration of two-level TPPC. This argument is based on several facts: First, the latency of unidirectional Viterbi is half that of symbol BCJR. Second, syndrome mlLLR generation, EPCC decoding and bit-LLR generation are all implemented tensor-symbol by tensor-symbol, achieving full parallelism on the tensor-symbol level. Third, it is only when QLDPC finds a syndrome error that EPCC decoding is turned on. Finally, branch metric computation in Viterbi and (5.5) is

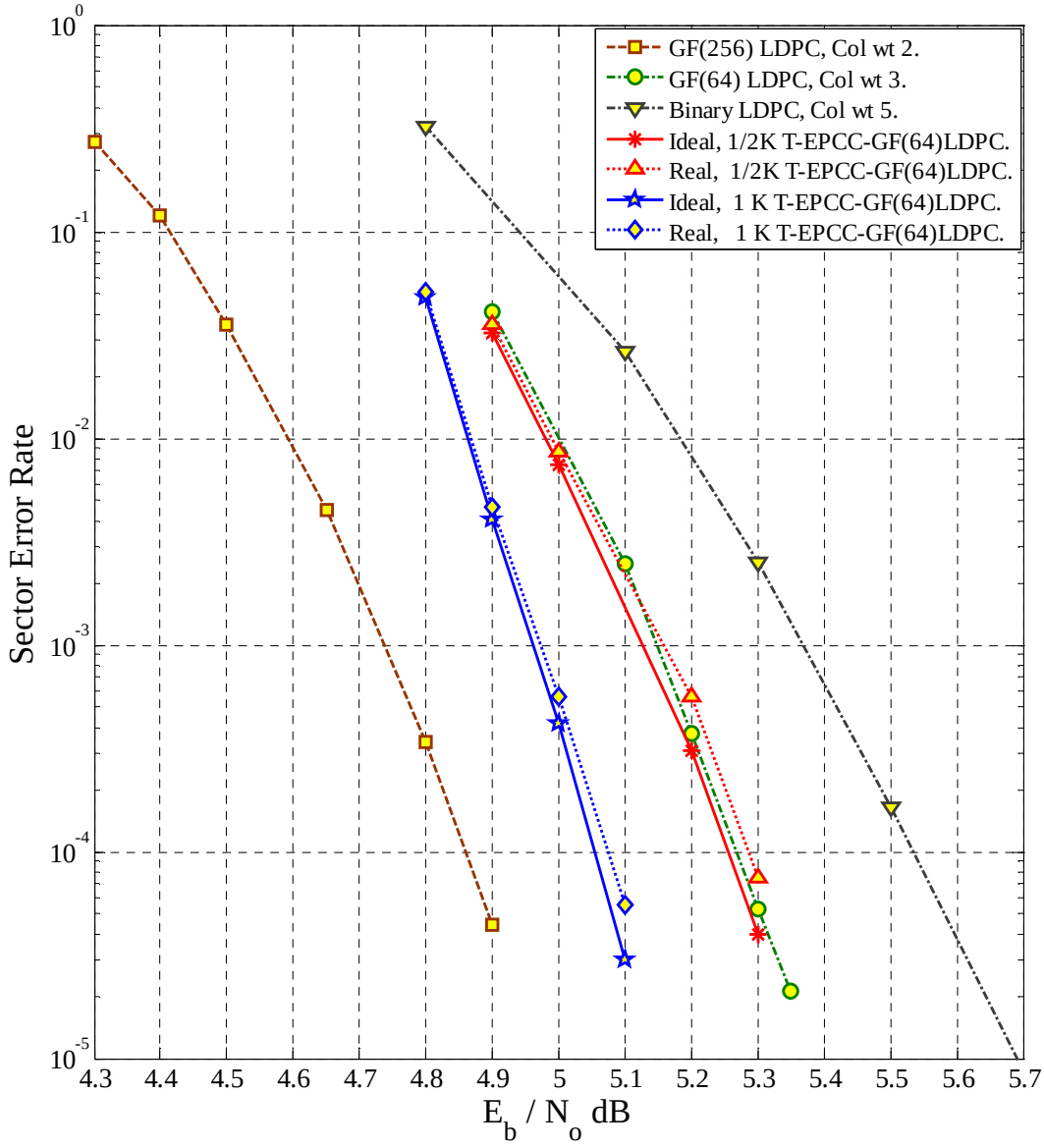


Figure 5.5: Comparing SER in environment of rate penalty$10\log_{10} 1/R$: 10×50 iterations of binary LDPC, 0×50 iterations of $GF(2^8)$ LDPC of column weight 2, 0×50 iterations of $GF(2^6)$ LDPC of column weight 3, and 3×50 iterations of practical $1/2$KB T-EPCC-$GF(2^6)$LDPC+RS($t = 6$), and 1KB T-EPCC-$GF(2^6)$LDPC+RS($t = 12$), both based on column weight 3 LDPC.

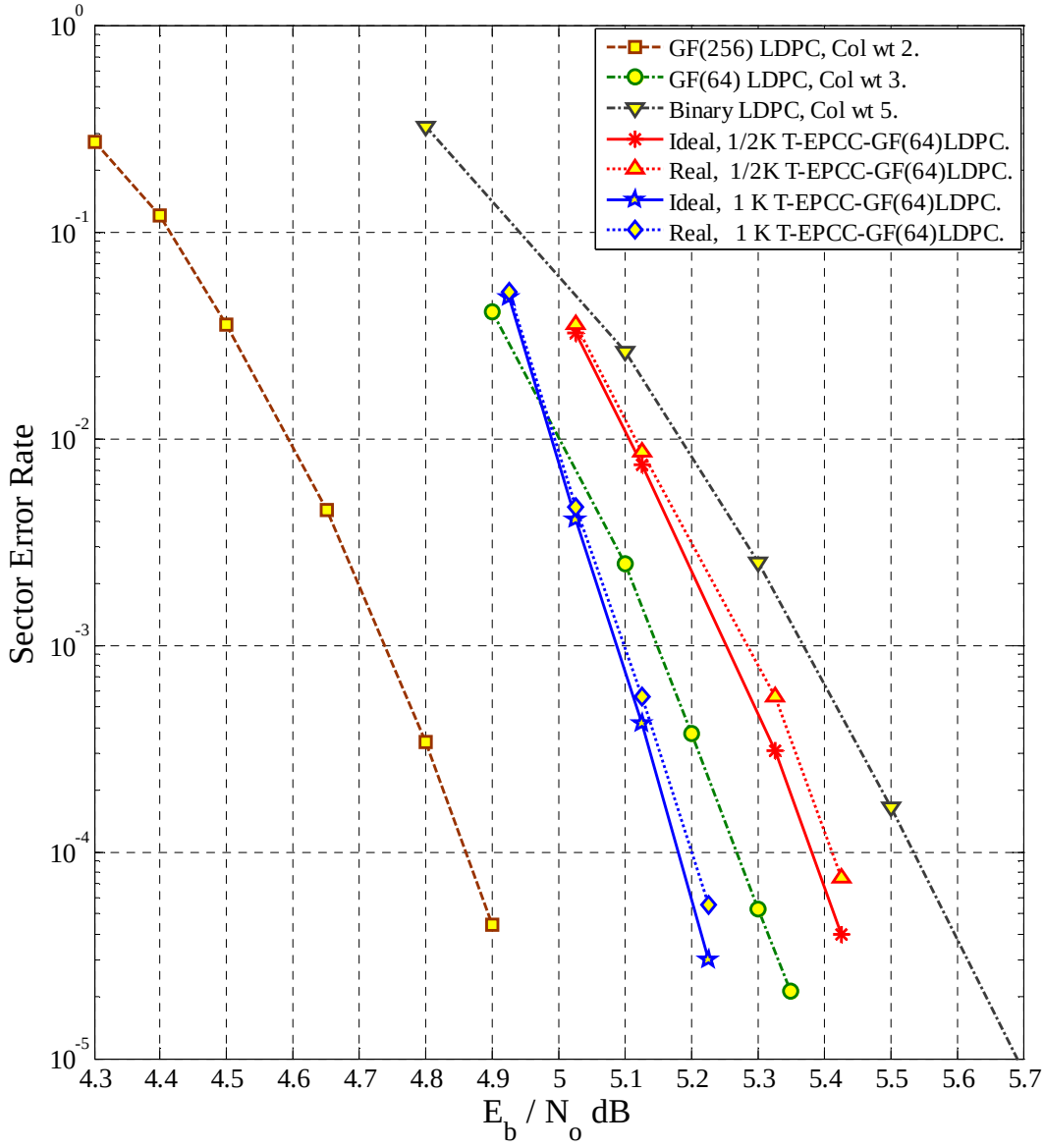


Figure 5.6: Comparing SER in environment of rate penalty$10\log_{10}1/R^2$: 10×50 iterations of binary LDPC, 0×50 iterations of $GF(2^8)$ LDPC of column weight 2, 0×50 iterations of $GF(2^6)$ LDPC of column weight 3, and 3×50 iterations of practical $1/2$KB T-EPCC-$GF(2^6)$LDPC+RS($t=6$), and 1KB T-EPCC-$GF(2^6)$LDPC+RS($t=12$), both based on column weight 3 LDPC.

only required at the first pass. For all subsequent iterations, however, only the *a priori* bias is updated in the second term of (5.5), and the branch update of Viterbi [79]. On the other hand, symbol-BCJR is the bottleneck for single level $GF(2^6)$-LDPC, where if we were to perform an extra global iteration, all the forward and backward parameters have to be recomputed in a serial fashion, this making back to back decoding hard to implement.

One very important feature of the TPPC setup, that single-level LDPC lacks, is its robustness to boundary error events. The presence of a syndrome-constraint means that errors spanning boundaries are broken by EPCC when attempting to independently satisfy the adjacent tensor symbol syndromes, then, in the next turbo iteration adjacent tensor-symbols are decorrelated. This mechanism enables TPPC to recover from these errors by iterative decoding. However, for errors with a zero error-syndrome which go undetected by EPCC, outer RS protection becomes handy.

Based on the fact that TPPC enables an increase in the redundancy of its component LDPC, in addition to simulation results demonstrating the utility of such lowered rate in combating the harsher compressed channel, we conjecture that if the sector length of both TPPC and single level systems is driven to infinity, TPPC will achieve strict error rate gains. This is mainly because of its surplus of redundancy compared to the single level code, whereas channel conditions and EPCC correction power do not change with replication of tensor symbols. Therefore, within a channel-capacity achieving argument, in the limit of infinite codeword length, we take the view that TPPC will bridge the gap to capacity further than any single level system could. Moreover, the advantage of TPPC for larger sector sizes is more timely than ever as the industry moves to the larger 4KB sector format [78].

Chapter 6

Conclusion

6.1 Summary

In this thesis, we have studied the BER of the serial concatenation of EPCC and interleaved RSCC over ISI channels as an alternative to a single interleaved RSCC over precoded and unprecoded ISI channels. The proposed TE-EPCC represents an enhancement to standard TE, in which a small portion of the outer convolutional code redundancy is reallocated to an inner high rate EPCC code that is tailored to the channel. The inner EPCC code attempts to correct any single dominant error pattern and a considerable portion of their multiple occurrences. To facilitate the study of system performance for a wide range of coding rates, interleaver sizes, and EPCC design parameters, we have derived an approximate upper bound on the BER of TE-EPCC that is easy to evaluate and that scales well with system parameters. We have also shown how EPCC enhances TE performance by reducing the frequency of error words of low Euclidean distance, which dominate the BER both in the cliff and error floor regions. Numerical results, calculated via the derived bound, indicate that TE-EPCC delivers substantial gain for short interleaver lengths compared to precoded and unprecoded TE, which makes it more attractive than conventional TE for hardware implementation. Also, we have demonstrated that for the Dicode ISI channel corrupted

by AWGN, TE-EPCC furnishes a uniform gain of 1.5 dB for puncturing rates above 0.75, which makes it suitable for high rate applications, such as magnetic and optical recording applications, while precoded TE is a better choice for lower coding rates. As a design guideline, we can conclude from the analytic BER study of TE-EPCC that a single EPCC subcode per an RSCC interleave of short length achieves a better performance-complexity tradeoff compared to $L > 1$ EPCC subcodes and a long interleave. This statement takes into consideration the savings in memory requirement of short-interleaver based TE compared to a system with a larger interleaver size, and the fact that TE-EPCC posses a large interleaver gain exponent for low Euclidean weight errors. Hence, short interleaved TE-EPCC of $L = 1$ is a good candidate for the implementation of a pipelined turbo decoder architecture.

Furthermore, we have studied the performance of TE-EPCC in perpendicular magnetic recording channels, where we have modified the EPCC list decoder to handle data dependent media noise. An improved concatenation scheme based on the turbo product concept was also proposed. A simulation study of these new coding systems shows that for a PMR read channel with normalized channel density of 1.35, 90% media noise, and a PRML target of $1 + 0.9D$, a rate $8/9$ TE-EPCC with 15 turbo iterations is 1.3 dB better than conventional rate-$8/9$ turbo equalization at BER 10^{-6}. On the other hand, a more complex, rate $8/9$ TP-EPCC with 5×15 iterations is 1.5 dB better than conventional rate-$8/9$ turbo equalization at the same BER. Also, iterative systems based on EPCC have reasonable burst statistics that are within the capability of outer RS ECC. However, the burstiness of the errors in failed sectors seems worse than conventional TE, pointing to a need for further investigation. Still, the proposed EPCC-based turbo equalizer does show a BER advantage over conventional TE in the bit error rate region of interest. Then, with RS sector interleaving, we can preserve more of the BER iterative gain after ECC. Moreover, The BER performance of the proposed EPCC-enhanced TE is comparable to that of LDPC, while the number of corrupted symbols is much smaller than LDPC when sectors fail.

In this thesis, we also have demonstrated how, in a tensor product concatenation, codes of short codeword length and low rate can be combined into high rate codes of nice algebraic properties. In addition, we showed that encoding of tensor product codes is linear time if the component codes are linear time encodable. We also demonstrated how the rate of channel matched EPCC can be substantially increased, while EPCC codeword length is kept within the order of the outer ECC symbol size, by tensor concatenation with a strong RS or LDPC of short codeword length. The tensor concatenation technique puts more pressure on the syndrome-correcting component code, wether RS or QLDPC. This is due to the compression of tensor-symbols into shorter syndrome symbols, concentrating the channel noise, which was equivalently spread over many tensor-symbol bits, into fewer syndrome-bits. Nevertheless, through a fully analytic multinomial SER estimation technique, we proved that the SER of the tensor component RS, assuming no component EPCC imperfections, outperforms single-level RS for a wide range of coding rates and sector sizes, where the gain is maximized for higher rates and larger sector. Motivated by the encouraging analytic study of T-EPCC, we proposed a practical decoder for T-EPCC-QLDPC based on the soft decoders of the constituent decoders. In the design of the practical soft iterative decoder of T-EPCC-QLDPC, we incorporated an outer RS code of low correction power to clean out the residual errors of T-EPCC-QLDPC and to provide a stopping criterion for the iterative decoder. Moreover, if the number of residual error after EPCC is larger than the outer RS correction power, erasure decoding of the outer RS is turned on, doubling the correction power compared to error-based decoding.

Simulation results show that T-EPCC-QLDPC achieves similar performance to single-level QLDPC with a $1/2$KB sector at 50% reduction in decoding complexity. Moreover, 1KB T-EPCC-QLDPC surpasses the performance of $1/2$KB single-level QLDPC at the same decoder complexity. In conclusion, we have established T-EPCC-QLDPC as a reasonable complexity approach to introducing non-binary LDPC to the perpendicular recording read channel architecture, paving the way to reliable higher recording densities.

6.2 Future work

One direction of future research is the design of T-EPCC-BLDPC codes, where one can study the performance of tensor concatenation based on binary rather than Q-ary LDPC and develop a method to generate bit-level LLR for the bit-syndromes (compared to symbol-level LLR for the symbol-syndromes) using a "Pyndiah-like" approach [49]. Moreover, in a similar fashion, one can design variable complexity decoders based on a given EPCC code. For example, considering an EPCC with say 6 parity bits, we can investigate the performance and complexity of using binary, $GF(4)$, $GF(8)$, $GF(16)$,$GF(32)$, and $GF(64)$ LDPC codes as component codes, where the use of a Pyndiah-like averaging method would allow for this design flexibility. In this method, p-bits are grouped and fixed at a certain position, then, the mlLLR of this $GF(2^p)$ position is calculated by averaging across classes corresponding to different values of the grouping, where the current position is kept fixed and all other locations are changed within each class.

Another direction of investigation is based on the fact that the TPPC construction allows the incorporation of short syndrome-correcting codes of low rate. Hence, we can study T-EPCC-RS with soft decoding of the component RS, the motive being that soft decoding is more effective for short low rate RS codes, as is demonstrated for example in [80].

One more area to investigate, is lowering the decoding complexity of the component QLDPC in T-EPCC-QLDPC. Here, the sparsity of the BP decoder messages stemming directly from the way syndrome mlLLRs are construed, reduces the performance loss of low complexity algorithms based on message truncation [81].

Bibliography

[1] H. Zhong, W. Xu, N. Xie, and T. Zhang, "Area-efficient min-sum decoder design for high-rate quasi-cyclic low-density parity-check codes in magnetic recording," *Magnetics, IEEE Transactions on*, vol. 43, no. 12, pp. 4117–4122, Dec. 2007.

[2] H. Zhong, T. Zhong, and E. F. Haratsch, "Quasi-cyclic LDPC codes for the magnetic recording channel: Code design and VLSI implementation," *Magnetics, IEEE Transactions on*, vol. 43, no. 3, pp. 1118–1123, March 2007.

[3] N. Varnica and A. Kavcic, "Optimized low-density parity-check codes for partial response channels," *Communications Letters, IEEE*, vol. 7, no. 4, pp. 168–170, April 2003.

[4] S. Sankaranarayanan, B. Vasic, and E. Kurtas, "Irregular low-density parity-check codes: Construction and performance on perpendicular magnetic recording channels," *Magnetics, IEEE Transactions on*, vol. 39, no. 5, pp. 2567–2569, Sept. 2003.

[5] B. Vasic and O. Milenkovic, "Combinatorial constructions of low-density parity-check codes for iterative decoding," *Information Theory, IEEE Transactions on*, vol. 50, no. 6, pp. 1156–1176, June 2004.

[6] G. Liva, W. Ryan, and M. Chiani, "Quasi-cyclic generalized LDPC codes with low error floors," *Communications, IEEE Transactions on*, vol. 56, no. 1, pp. 49–57, January 2008.

[7] M. Yang, W. Ryan, and Y. Li, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *Communications, IEEE Transactions on*, vol. 52, no. 4, pp. 564–571, April 2004.

[8] Z. Li, L. Chen, L. Zeng, S. Lin, and W. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *Communications, IEEE Transactions on*, vol. 54, no. 1, pp. 71–81, Jan. 2006.

[9] Y. Han and W. Ryan, "Concatenating a structured LDPC code and a constrained code to preserve soft-decoding, structure, and burst correction," *Magnetics, IEEE Transactions on*, vol. 42, no. 10, pp. 2558–2560, Oct. 2006.

[10] B. Kurkoski, P. Siegel, and J. Wolf, "Joint message-passing decoding of LDPC codes and partial-response channels," *Information Theory, IEEE Transactions on*, vol. 48, no. 6, pp. 1410–1422, Jun 2002.

[11] Y. Han and W. Ryan, "LDPC decoder strategies for achieving low error floors," *Information Theory and Applications Workshop, 2008*, pp. 277–286, 27 2008-Feb. 1 2008.

[12] R. Cideciyan, E. Eleftheriou, and T. Mittelholzer, "Perpendicular and longitudinal recording: A signal-processing and coding perspective," *Magnetics, IEEE Transactions on*, vol. 38, no. 4, pp. 1698–1704, Jul 2002.

[13] H. Song, R. Todd, and J. Cruz, "Applications of low-density parity-check codes to magnetic recording channels," *Selected Areas in Communications, IEEE Journal on*, vol. 19, no. 5, pp. 918–923, May 2001.

[14] X. Hu and B. V. K. V. Kumar, "Evaluation of low-density parity-check codes on perpendicular magnetic recording model," *Magnetics, IEEE Transactions on*, vol. 43, no. 2, pp. 727–732, Feb. 2007.

[15] P. Chaichanavong and G. Burd, "On the concatenation of soft inner code with Reed-Solomon code for perpendicular magnetic recording," *Magnetics, IEEE Transactions on*, vol. 43, no. 2, pp. 744–749, Feb. 2007.

[16] Z. Keirn, V. Krachkovsky, E. Haratsch, and H. Burger, "Use of redundant bits for magnetic recording: single-Parity codes and Reed-Solomon error-correcting code," *Magnetics, IEEE Transactions on*, vol. 40, no. 1, pp. 225–230, Jan. 2004.

[17] V. Dorfman and J. Wolf, "Postprocessing using a single-parity interleaved block code for a Viterbi detector," *Magnetics, IEEE Transactions on*, vol. 40, no. 4, pp. 3090–3092, July 2004.

[18] K. Saeki and Z. Keirn, "Optimal combination of detection and error correction coding for magnetic recording," *Magnetics, IEEE Transactions on*, vol. 37, no. 2, pp. 708–713, Mar 2001.

[19] W. Bliss, R. Karabed, K. Zhang, C. Varanasi, and J. Ashley, "Sector error rate estimation of concatenated codes in magnetic recording," *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 9, pp. 2726–2730, 2001.

[20] W. Feng, A. Vityaev, G. Burd, and N. Nazari, "On the performance of parity codes in magnetic recording systems," *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, vol. 3, pp. 1877–1881 vol.3, 2000.

[21] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. of IEEE ICC*, Geneva, Switzerland, May 1993, pp. 1740–1745.

[22] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. on Inform. Theory*, vol. 42, pp. 409–428, March 1996.

[23] P. H. Siegel, D. Divsalar, E. Eleftheriou, J. Hagenauer, and D. Rowitch, "The turbo principle, from theory to practice I," *IEEE Journal on Sel. Areas in Communi.*, vol. 19, no. 5, pp. 793–799, May 2001.

[24] ——, "The turbo principle, from theory to practice II," *IEEE Journal on Sel. Areas in Communi.*, vol. 19, no. 9, pp. 1657–1661, September 2001.

[25] C. Douillard, M. Jzquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *Eur. Trans. Telecommun.*, vol. 6, pp. 507–511, Sep./Oct. 1995.

[26] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. on Communi.*, vol. 50, no. 5, pp. 754–767, May 2002.

[27] J. Li, K. Narayanan, E. Kurtas, and C. Georghiades, "On the performance of high-rate TPC/SPC codes and LDPC codes over partial response channels," *Communications, IEEE Transactions on*, vol. 50, no. 5, pp. 723–734, May 2002.

[28] J. Li, E. Kurtas, K. Narayanan, and C. Georghiades, "Iterative decoding of turbo product codes over PR-equalized lorentzian channels with colored noise," *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 5, pp. 2972–2976 vol.5, 2001.

[29] W. Ryan, "Performance of high rate turbo codes on a PR4-equalized magnetic recording channel," *Communications, 1998. ICC 98. Conference Record.1998 IEEE International Conference on*, vol. 2, pp. 947–951 vol.2, Jun 1998.

[30] M. Reed, C. Schlegel, P. Alexander, and J. Asenstorfer, "Iterative multiuser detection for CDMA with FEC: Near-single-user performance," *Communications, IEEE Transactions on*, vol. 46, no. 12, pp. 1693–1699, Dec 1998.

[31] T. V. Souvignier, M. Oberg, P. H. Siegel, R. E. Swanson, and J. K. Wolf, "Turbo decoding for partial response channels," *IEEE Trans. on Communi.*, vol. 48, no. 8, pp. 1297–1308, August 2000.

[32] L. McPheters, S. McLaughlin, and K. Narayanan, "Precoded PRML, serial concatenation, and iterative (turbo) decoding for digital magnetic recording," *Magnetics, IEEE Transactions on*, vol. 35, no. 5, pp. 2325–2327, Sep 1999.

[33] K. Narayanan, "Effect of precoding on the convergence of turbo equalization for partial response channels," *Selected Areas in Communications, IEEE Journal on*, vol. 19, no. 4, pp. 686–698, Apr 2001.

[34] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *Information Theory, IEEE Transactions on*, vol. 44, no. 3, pp. 909–926, May 1998.

[35] M. Oberg and P. H. Siegel, "Performance analysis of turbo-equalized partial response channels," *IEEE Trans. on Communi.*, vol. 49, no. 3, pp. 436–444, March 2001.

[36] J. Moon and J. Park, "Detection of prescribed error events: Application to perpendicular recording," *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, vol. 3, pp. 2057–2062, May 2005.

[37] J. Park and J. Moon, "High-rate error-correction codes targeting dominant error patterns," *Magnetics, IEEE Transactions on*, vol. 42, no. 10, pp. 2573–2575, Oct. 2006.

[38] ——, "A new class of error-pattern-correcting codes capable of handling multiple error occurrences," *Magnetics, IEEE Transactions on*, vol. 43, no. 6, pp. 2268–2270, June 2007.

[39] H. Alhussien, J. Park, and J. Moon, "Iterative decoding based on error pattern correction," *Magnetics, IEEE Transactions on*, vol. 44, no. 1, pp. 181–186, Jan. 2008.

[40] J. Park and J. Moon, "Error-pattern-correcting cyclic codes tailored to a prescribed set of error cluster patterns," *IEEE Trans. on Inform. Theory*, to appear 2009.

[41] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *Information Theory, IEEE Transactions on*, vol. 18, no. 1, pp. 170–182, Jan 1972.

[42] J. Forney, G., "Generalized minimum distance decoding," *Information Theory, IEEE Transactions on*, vol. 12, no. 2, pp. 125–131, Apr 1966.

[43] J. Wolf, "On codes derivable from the tensor product of check matrices," *Information Theory, IEEE Transactions on*, vol. 11, no. 2, pp. 281–284, Apr 1965.

[44] J. Wolf and B. Elspas, "Error-locating codes–a new concept in error control," *Information Theory, IEEE Transactions on*, vol. 9, no. 2, pp. 113–117, Apr 1963.

[45] A. Fahrner, H. Grießer, R. Klarer, and V. Zyablov, "Low-complexity GEL codes for digital magnetic storage systems," *Magnetics, IEEE Transactions on*, vol. 40, no. 4, pp. 3093–3095, July 2004.

[46] H. Imai and H. Fujiya, "Generalized tensor product codes," *Information Theory, IEEE Transactions on*, vol. 27, no. 2, pp. 181–187, Mar 1981.

[47] P. Chaichanavong and P. Siegel, "Tensor-product parity code for magnetic recording," *Magnetics, IEEE Transactions on*, vol. 42, no. 2, pp. 350–352, Feb. 2006.

[48] ——, "Tensor-product parity codes: Combination with constrained codes and application to perpendicular recording," *Magnetics, IEEE Transactions on*, vol. 42, no. 2, pp. 214–219, Feb. 2006.

[49] R. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *Communications, IEEE Transactions on*, vol. 46, no. 8, pp. 1003–1010, Aug 1998.

[50] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *Information Theory, IEEE Transactions on*, vol. 20, no. 2, pp. 284–287, Mar 1974.

[51] J. Caroselli, S. Altekar, P. McEwen, and J. Wolf, "Improved detection for magnetic recording systems with media noise," *Magnetics, IEEE Transactions on*, vol. 33, no. 5, pp. 2779–2781, Sep 1997.

[52] J. Moon and J. Park, "Pattern-dependent noise prediction in signal-dependent noise," *Selected Areas in Communications, IEEE Journal on*, vol. 19, no. 4, pp. 730–743, Apr. 2001.

[53] D. N. Rowitch, "Convolutional and turbo coded multicarrier direct sequence CDMA, and applications of turbo codes to hybrid ARQ communication systems," Ph.D. dissertation, Univ. California, San Diego, La Jolla, June 1998.

[54] M. Oberg and P. H. Siegel, "Performance bound for parity-check coded partial-response channels," in *Proc. of IEEE ICC*, vol. 9, 2001, pp. 2701–2705.

[55] G. Ungerboeck, "Channel coding with multilevel/phase signals," *Information Theory, IEEE Transactions on*, vol. 28, no. 1, pp. 55–67, Jan 1982.

[56] ——, "Trellis-coded modulation with redundant signal sets Part I: Introduction," *Communications Magazine, IEEE*, vol. 25, no. 2, pp. 5–11, Feb 1987.

[57] ——, "Trellis-coded modulation with redundant signal sets Part II: State of the art," *Communications Magazine, IEEE*, vol. 25, no. 2, pp. 12–21, Feb 1987.

[58] J. Moon, "Discrete-time modeling of transition noise dominant channels and study of detection performance," *Magnetics, IEEE Transactions on*, vol. 27, no. 6, pp. 4573–4578, Nov 1991.

[59] H. Sawaguchi, Y. Nishida, H. Takano, and H. Aoi, "Performance analysis of modified PRML channels for perpendicular recording systems," *Journal of Magnetism and Magnetic Materials*, vol. 235, pp. 265–272, Nov.

[60] M. Madden, M. Oberg, Z. Wu, and R. He, "Read channel for perpendicular magnetic recording," *Magnetics, IEEE Transactions on*, vol. 40, no. 1, pp. 241–246, Jan. 2004.

[61] J. Moon, "Signal-to-noise ratio definition for magnetic recording channels with transition noise," *Magnetics, IEEE Transactions on*, vol. 36, no. 5, pp. 3881–3883, Sep 2000.

[62] H. Thapar and A. Patel, "A class of partial response systems for increasing storage density in magnetic recording," *Magnetics, IEEE Transactions on*, vol. 23, no. 5, pp. 3666–3668, Sep 1987.

[63] J. Moon and L. R. Carley, "Performance comparison of detection methods in magnetic recording," *IEEE Trans. on Magn.*, vol. 26, no. 6, pp. 3155–3172, November 1990.

[64] J. Moon and W. Zeng, "Equalization for maximum likelihood detectors," *IEEE Trans. on Magn.*, vol. 31, no. 2, pp. 1083–1088, March 1995.

[65] A. Kavcic and J. Moura, "Correlation-sensitive adaptive sequence detection," *Magnetics, IEEE Transactions on*, vol. 34, no. 3, pp. 763–771, May 1998.

[66] D. Sridhara, T. Fuja, and R. M. Tanner, "Low density parity check codes from permutation matrices," *Conference on Information Sciences and Systems, The Johns Hopkins University*, March 21-23 2001.

[67] R. Tanner, D. Sridhara, A. Sridharan, T. Fuja, and J. Costello, D.J., "LDPC block and convolutional codes based on circulant matrices," *Information Theory, IEEE Transactions on*, vol. 50, no. 12, pp. 2966–2984, Dec. 2004.

[68] Z. Zhang, T. Duman, and E. Kurtas, "Information rates of binary-input intersymbol interference channels with signal-dependent media noise," *Magnetics, IEEE Transactions on*, vol. 39, no. 1, pp. 599–607, Jan 2003.

[69] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. on Inform. Theory*, vol. 42, no. 2, pp. 429–445, March 1996.

[70] J. Massey, "Shift-register synthesis and BCH decoding," *Information Theory, IEEE Transactions on*, vol. 15, no. 1, pp. 122–127, Jan 1969.

[71] J. Wolf, "ECC performance of interleaved RS codes with burst errors," *Magnetics, IEEE Transactions on*, vol. 34, no. 1, pp. 75–79, Jan 1998.

[72] B. Kurkoski, K. Yamaguchi, and K. Kobayashi, "Turbo equalization with single-parity check codes and unequal error protection codes," *Magnetics, IEEE Transactions on*, vol. 42, no. 10, pp. 2579–2581, Oct. 2006.

[73] M. Davey and D. MacKay, "Low-density parity check codes over GF(q)," *Communications Letters, IEEE*, vol. 2, no. 6, pp. 165–167, Jun 1998.

[74] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *Information Theory, IEEE Transactions on*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[75] Z. Li and B. Kumar, "A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph," *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, vol. 2, pp. 1990–1994, Nov. 2004.

[76] H. Song and J. Cruz, "Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording," *Magnetics, IEEE Transactions on*, vol. 39, no. 2, pp. 1081–1087, Mar 2003.

[77] D. Arnold and H.-A. Loeliger, "On the information rate of binary-input channels with memory," *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 9, pp. 2692–2695 vol.9, 2001.

[78] W. Chang and J. Cruz, "Nonbinary LDPC codes for 4-kB sectors," *Magnetics, IEEE Transactions on*, vol. 44, no. 11, pp. 3781–3784, Nov. 2008.

[79] W. Chang and J. R. Cruz, "Comments on "Performance and decoding complexity of nonbinary LDPC codes for magnetic recording" [Jan 08 211-216]," *Magnetics, IEEE Transactions on*, vol. 44, no. 10, pp. 2423–2424, Oct. 2008.

[80] J. Bellorado, A. Kavcic, and L. Ping, "Soft-input, iterative, Reed-Solomon decoding using redundant parity-check equations," *Information Theory Workshop, 2007. ITW '07. IEEE*, pp. 138–143, Sept. 2007.

[81] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity, low-memory EMS algorithm for non-binary LDPC codes," *Communications, 2007. ICC '07. IEEE International Conference on*, pp. 671–676, June 2007.