Channel Coding for High Speed Links

by

Natasa Blitvic

B. Sc. A. Electrical Engineering, University of Ottawa, 2005

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

February, 2008

Author .....
Department of Electrical Engineering and Computer Science
December 7, 2007

Certified by .....
Vladimir Stojanović, Assistant Professor of Electrical Engineering
M.I.T. Thesis Supervisor

Accepted by .....
Terry P. Orlando
Professor of Electrical Engineering
Chairman, Department Committee on Graduate Theses

Channel Coding for High Speed Links

by

Natasa Blitvic
B. Sc. A. Electrical Engineering, University of Ottawa, 2005

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology
December, 2007

Abstract

This thesis explores the benefit of channel coding for high-speed backplane or chip-to-chip interconnects, referred to as the high-speed links. Although both power-constrained and bandwidth-limited, the high-speed links need to support data rates in the Gbps range at low error probabilities. Modeling the high-speed link as a communication system with noise and intersymbol interference (ISI), this work identifies three operating regimes based on the underlying dominant error mechanisms. The resulting framework is used to identify the conditions under which standard error control codes perform optimally, incur an impractically large overhead, or provide the optimal performance in the form of a single parity check code. For the regime where the standard error control codes are impractical, this thesis introduces low-complexity block codes, termed pattern-eliminating codes (PEC), which achieve a potentially large performance improvement over channels with residual ISI. The codes are systematic, require no decoding and allow for simple encoding. They can also be additionally endowed with a $(0, n-1)$ run-length-limiting property. The simulation results show that the simplest PEC can provide error-rate reductions of several orders of magnitude, even with rate penalty taken into account. It is also shown that channel conditioning, such as equalization, can have a large effect on the code performance and potentially large gains can be derived from optimizing the equalizer jointly with a pattern-eliminating code. Although the performance of a pattern-eliminating code is given by a closed-form expression, the channel memory and the low error rates of interest render accurate simulation of standard error-correcting codes impractical. This work proposes performance estimation techniques for coded high-speed links, based on the underlying regimes of operation. It also introduces an efficient algorithm for computing accurate marginal probability distributions of signals in a coded high-speed link.

Thesis Supervisor : Vladimir Stojanovic
Title : Assistant Professor of Electrical Engineering

Acknowledgement

The author would like to thank

Professor Vladimir Stojanovic, for his time, support, advice, and for painstakingly making her understand the meaning of *practical*. His breadth of knowledge and diversity of interests allowed this work on high-speed links to evolve in an admittedly unorthodox direction.

Professor Lizhong Zheng, for his dedicated interest in the project and enlivening discussions.

The Integrated Systems Group at MIT, for their friendship and skillful advice. In particular, the author is grateful to Maxine Lee, for paving the way; Ranko Sredojevic and Yan Li for their help with optimization; Fred Chen, Sanquan Song and Byungsub Kim for their knowledge of circuit design and high-speed links. This project did not have much in common with the work of Ajay Joshi, Ben Moss and Olivier Bichler, but what would life be without friends.

The author would also like to thank

Her parents, for **everything**,

and

Vicente.

Contents

Introduction

This thesis explores the benefit of channel coding for high-speed backplane or chip-to-chip interconnects, commonly referred to as the *high-speed links*. High-speed links are ubiquitous in modern computing and routing. In a personal computer, for instance, they link the central processing unit to the memory, while in backbone routers thousands of such links interface to the crossbar switch. Arising from the critical tasks they are designed to perform, high-speed links are subject to stringent throughput, accuracy and power consumption requirements. A typical high-speed link operates at a data rate on the order of 10 Gbps with error probability of approximately 10^{-15}.

Given the bandwidth-limited nature of the backplane communication channel, the ever-increasing demands in computing and routing speeds place a large burden on high-speed links. Specifically, high data rates exacerbate the inter-symbol interference (ISI), while the power constraints and the resulting complexity constraints limit the ability to combat the ISI. As a result, the system is no longer able to provide the required quality of communication. In fact, this residual ISI limits the achievable link data rates to an order of magnitude below their projected capacity [5].

Although most modern communication systems employ some form of coding as a technique to improve the quality of communication, the residual ISI severely impairs the performance of many such techniques. This is demonstrated in a thesis by M. Lee [4], which consists of a series of experimental results that evaluate the potential of standard error-correction and error-detection schemes for high-speed link applications. Moreover, coding schemes are also generally considered impractical for high-speed links due to the required overhead and the resulting rate penalty. Thus, most research efforts to bring more advanced communication techniques to high-speed

links have focused on improved signaling/modulation techniques and equalization [5, 16]. However, due to the previous lack of adequate theoretical and simulation frameworks, the previous conclusions regarding the benefit of coding for high-speed links are based on incomplete results and therefore remain speculative. Specifically, Monte-Carlo-based simulation techniques are not suitable for performance estimation of a coded high-speed link due to the low error probabilities, while the closed-form expressions pertain to the limiting cases and may therefore not be sufficiently accurate. As a result, the most significant body of work on coding for high-speed links, [4], is purely experimental. The resulting practical constraints limit the depth of the analysis in [4] and further constrain the scope of the study to standard coding techniques, which may not be optimal in the high-speed link setting.

The present work addresses the issue of coding for high-speed links from a theoretical perspective, by abstracting the high-speed link as a general system with noise and ISI. This abstraction allows for a classification of possible error mechanisms, which enables both a deeper characterization of the behavior of different codes in a high-speed link and the development of new coding techniques adapted to these systems. The benefits of the regime classification also extend to system simulation, by allowing for more efficient simulation methods tailored for different regimes. In particular, this thesis develops the following results:

- A more complete characterization of error mechanisms occurring in a high-speed link. This enables the classification of system's operating conditions based on the dominant error mechanism as one of three possible regimes, namely, the large-noise, the worst-case-dominant and the large-set-dominant regimes.

- A deeper characterization of codes for high-speed links. For instance, a characterization of error mechanisms allows to identify the conditions under which standard error control codes incur little or no performance impairements due to the residual ISI. Note that these include, *but are not limited to*, the cases where the ISI is relatively weak compared to the noise.

- A new approach to coding under the worst-case-dominant regime where error is principally due to the occurrences of certain symbol patterns. The resulting codes, termed the *pattern-eliminating codes* provide the benefit of low complexity, allowing for simple encoding and requiring virtually no decoding.

- A theoretical framework for interpreting previously-documented experimental behaviors, such as [4] or [11, 12].

- New simulation methods for coded or uncoded high-speed links. The regime classification provides a more accurate guideline for biasing the system parameters in simulation to capture error behaviors at low probabilities. Moreover, computational approaches that enable performance estimation without parameter biasing are identified for each of the regimes.

- An efficient numerical algorithm for computing marginal probability distributions in a coded system. The algorithm is of particular use under operating conditions that render the previous techniques impractical, either due to excessive computational complexity or insufficient accuracy. Note that, since the performance of pattern-eliminating codes is given by a closed-form expression for all regimes of interest, the algorithm focuses on systematic linear block codes whose behavior is the focus of previous experimental work.

The three operating regimes are described qualitatively in Chapter 1 and quantitatively in Chapter 2. The main body of results on coding for systems with noise and ISI, and the subsequent specialization to high-speed links, is contained in Chapter 2. The issue of system simulation for high-speed links is addressed in Chapter 3.

Chapter 1

The High-speed Link: The Reality and the Abstraction

High-speed links typically refer to backplane or chip-to-chip interconnects that operate at very high data rates ($\sim$ 10 Gbps), low bit-error rate ($\sim 10^{-15}$) and with high energy efficiency. These stringent requirements arise from the critical tasks that high-speed links are designed to perform, as well as the global die-level and system-level power constraints. In a personal computer, for instance, they link the central processing unit to the memory, where ensuring an adequate speed and accuracy of the data transfer is essential. On a much larger scale, in backbone routers, thousands of such links interface to the crossbar switch. There, adequate accuracy guarantees the quality of the network traffic, the high speed reduces the number of links required to achieve the desired throughput, and high power efficiencies prevent that scaling from becoming prohibitive. In fact, the high-speed links form an integral part of many systems and their limitations thus generally have wide-ranging repercussions. This chapter begins by further describing the motivation behind the current high-speed-link research. Following the motivation, the high-speed link is re-introduced in a more abstracted form: that of a communication system with inter-symbol interference (ISI) and additive white Gaussian noise (AWGN). The payoff of this abstraction is realized in the last section, which introduces three interference-and-noise scenarios that provide the foundation for the development of the subsequent chapters.

1.1 Motivation

The increasing demands in network and processing speeds have placed a large burden on high-speed links, as it is becoming increasingly difficult to keep up with the increasing data rates while maintaining the same reliability and adequately low power consumption. Referring again to the part played by high-speed links in backbone routers, [4] cites the following relevant example. Consider the result of using current technology to create a 40 Tb/s crossbar router chip. Since cross-bar router switches currently support 1 Tb/s, striving to reach the 40 Tb/s mark is a realistic goal. To achieve that throughput, 4000 of the current 10 Gb/s transceivers would have to be deployed. Since each transceiver uses a differential pair, the switch chip would need 8000 I/O pins, thus requiring the total on-chip area of 4000 mm^2. The resulting switch card would be roughly 4 meters (160 inches) wide and 2.5 meters (100 inches) long. Furthermore, since each transceiver currently dissipates 40 mW/Gb/s, the power consumption for the crossbar chip would total 1.6 kW, an unrealistically large number.

The above example illustrates the need for improved data rates and energy efficiencies of high-speed links. However, at higher data rates, the bandwidth-limited nature of the backplane as a communication channel coupled with signal reflections off of impedance discontinuities results in a large amount of inter-symbol interference (ISI). In these conditions, clever circuit design is insufficient to maintain adequate speed and reliability of the communication. Furthermore, the rate at which the communication channel degrades with the increasing data rates, coupled with tight power constraints, renders adequate[1] channel equalization impractical. The resulting uncompensated ISI, as further discussed in the following sections, becomes one of the dominant error mechanisms and limits the achievable link data rates to an order of magnitude below their projected capacity [5]. It is thus necessary to probe deeper into the communication theory and rediscover, or develop, more energy-efficient methods of ensuring adequate communication in these conditions. This thesis explores the

[1] i.e. sufficient to achieve error rates on the order of 10^{-15}.

benefit of channel coding for high-speed links. In particular, it develops a new family of low-complexity codes, termed the *pattern-eliminating codes*, suitable in high-ISI regimes and provides a framework for simulation of standard error-control codes over channels with non-negligible ISI.

1.2 High-speed Link as a Communication System

The three principal error mechanisms in high-speed links, as described in [6] are *interference*, *noise*, and *timing jitter*. The interference in high-speed links is due to both the inter-symbol interference (ISI) and the crosstalk. The ISI is principally the result of the dispersion, due to the fundamental loss mechanisms in the wire, including skin effect and dielectric loss, and the signal reflections off of impedance discontinuities. Crosstalk, which can also be considered as co-channel interference (CCI), is due to multiple signals traveling through the same backplane. The effects of both the ISI and the CCI on the received signal can be described through the convolution of the channel impulse response, sampled at symbol times, with the history of the transmitted symbols. Noise in high-speed links is typically small with respect to the signal magnitude. It is principally ascribed to thermal noise and transistor device noise, and is therefore modeled as additive, white and Gaussian (AWGN) with standard deviation in the range of 3 mV down to 0.3 mV approximately, where the transmitted signal swing is $\pm$ 1V. The timing jitter is tied to the transmitter phase-locked loops (PLL) and the received clock data recovery (CDR) circuits. Various jitter models are available and many are overviewed in [6]. Since the ISI has been identified as the dominant error mechanism in a high-speed link [5], this thesis focuses on ISI-limited channels with AWGN.

The resulting simplified model of a typical high-speed link as a communication system is shown in Fig.1-1. The system employs PAM2 modulation and the equivalent communication channel is discrete. The latter also includes the effects of any signal processing in the transmitter or the receiver, such as equalization or matched-filtering. Due to the practical limits of equalization in high-speed links, the equalized channel

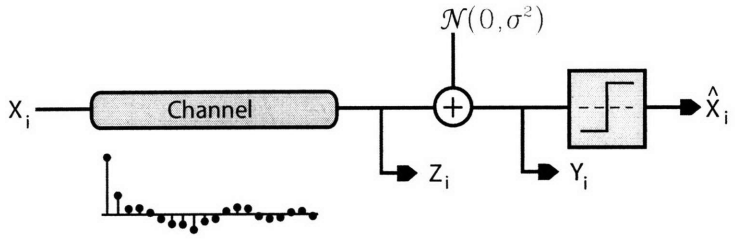

Figure 1-1: Model of a High-speed link as a Communication System

response typically contains some amount of *residual* inter-symbol interference. Due to hardware complexity constraints, the detection scheme at the receiver is a blind application of the symbol-by-symbol maximum a posteriori (MAP) detection [3] for channels without ISI, rather than a more complex sequence detection scheme, such as [26], developed for channels with ISI.

Concerning notation, it is convenient to denote the received signal at some time index i as Y_i and represent it as the sum of a signal component Z_i and the noise component N_i. Note that, throughout this thesis, upper-case notation is used to represent random variables, while the lower-case notation denotes the corresponding realizations and other deterministic quantities. The exception to this rule is the length of the channel response, denoted by L in order to be distinguished more easily from the ubiquitous time index i. Thus, for a channel of length L, the signal component Z_i is given as the sum of the current symbol and the $L-1$ previously[2] transmitted symbols, weighted by the channel coefficients. The value z such that when $Z = z$ the signal suffers no ISI is referred to as the *signal mean*, despite the fact that for constrained (coded) symbols z may not equal the expectation of the random variable Z_i.

Given a sequence of transmitted symbols $X_i, \ldots, X_{i-L+1} \in \{-1, 1\}$, where X_i is transmitted *last*, and some equivalent channel response of length L specified by the coefficients $h_0, \ldots, h_{L-1} \in \mathbb{R}$, the corresponding Y_i is given by

$$Y_i = \sum_{k=0}^{L-1} X_{i-k} h_k + N_i \tag{1.1}$$

[2]In case the channel also causes precursor ISI, that is, has a non-causal impulse response, the $L-1$ interfering symbols also contain "future" symbols.

Since, Y_i is discrete, its marginal probability distribution is specified by the probability mass function (PMF), denoted by f_{Y_i}, or alternatively, by the cumulative mass function (CMF), F_{Y_i}. For a communication channel with ISI, the dependencies between the received symbols resulting from the convolution equation (Eqn. 2.1) imply that the set of marginal distributions $f_{Y_i}, \ldots, f_{Y_{i+L-1}}$ is typically not sufficient to specify the joint distribution $f_{Y_i,\ldots,Y_{i+L-1}}$. However Chapter 2, among other results, describes some special circumstances under which the joint and the marginal distributions are effectively interchangeable.

1.3 Classification of Error Mechanisms in High-speed Links

Motivated by the model of a high-speed link introduced in the previous section, this section provides a more thorough characterization of the error mechanisms in systems limited by AWGN and ISI. Typically, the dominant error mechanism is loosely defined as the most likely source of detection errors. For instance, [6] observes that the inter-symbol interference, rather than noise alone or the timing jitter, is the dominant error mechanism in a high-speed link. In the present context, however, the term takes on a more precise meaning. Specifically, the dominant error mechanism refers to an *attribute of a set of interference events* which are found to be responsible for some large proportion of the detection errors.

The concept of a dominant error mechanism is formalized in Section 2.2 of Chapter 2. In the meanwhile, to gain a qualitative understanding of the concept, different error mechanisms are examined through the *a posteriori probability distribution* of the random variable Z_i, defined in the present context as the probability distribution of Z_i conditioned on the occurrence of an error event. More precisely, letting $P(Z_i = z)$ denote the a priori probability of observing some signal component z of the total received signal, the corresponding unilateral a posteriori probabilities are given by $P(Z_i = z \mid Y_i < 0, X_i = 1)$ and $P(Z_i = z \mid Y_i > 0, X_i = -1)$. In a

loose sense, the a posteriori distribution specifies the proportion of the errors that are due to each possible interference event. However, several factors jointly determine the a posteriori probability distributions and different combinations of these factors give rise to distinct *scenarios* or *regimes*. Specifically, it is clear that in a communication system with noise and ISI, the nature of the main error mechanism is some factor of the magnitude of the ISI relative to the transmitted symbol power, the noise variance relative to the transmitted symbol power, and the magnitude of the ISI relative to the noise variance. A more precise characterization yields the following three noise-and-interference scenarios, which enable the classification of any communication system that is principally limited by noise and ISI. They consist of the *large-noise scenario*, the *worst-case-dominant* scenario, and the *large-set-dominant* scenario. In the large-noise scenario, ISI is negligible with respect to noise and the latter dominates the error expression, while the error in the worst-case-dominant scenario is principally attributable to the occurrence of symbol patterns causing worst-case interference. The large-set-dominant scenario encompasses the remaining conditions, but also allows for a general result regarding joint error behaviors. The corresponding classification framework is at the core of the results formulated in the later chapters, which develop both codes and simulation methodologies to suit particular regimes in a system with noise and ISI.

Prior to discussing individual scenarios, note that, from the theoretical perspective, the three different scenarios represent three different limiting behaviors. This view is further discussed in Chapter 2. On the other hand, from a practical standpoint, the three scenarios provide a classification framework where the boundaries are context-dependent. For instance, regarding the performance of codes optimized for a given limiting behavior[3], the boundary of the corresponding regime is set to encompass the operating conditions under which such codes provide a benefit. Similarly, from the point of view of performance estimation, it is convenient to consider as large-set dominant all conditions under which the error events can be considered

[3]In Chapter 2, standard error correction codes are shown to perform optimally in the limit of the large noise regime and a subcase of the large-set-dominant regime, while the pattern-eliminating codes are developed for the limit of the worst-case-dominant regime.

as statistically independent, with some sufficient accuracy.

The following discussion pertains to an arbitrary real channel of length L whose smallest-magnitude coefficient is denoted by δ. The worst-case interference incurred by the signal, that is, the maximum deviation from the signal mean in either direction, is represented by some $\Delta > 0$. In other words, letting z denote the corresponding signal mean[4], the random variable Z_i takes values from the interval $[z - \Delta, z + \Delta]$. Also, note that the worst-case interference is lower-bounded by $(L-1)\delta$ and that possible values of Z_i occur in increments of at least 2δ. The quantity σ_{ISI}^2 denotes the variance of the ISI, that is, of the random variable $Z_i - h_0 X_i$. The noise (AWGN) is assumed to be independent of the signal, with zero mean and some variance σ^2.

The two sets of plots of Figures 1-2 and 1-3, depicting the a posteriori probability distributions for the random variable Z_i as a function of σ and δ, are used in the sections that follow to exemplify and link the three scenarios. The probability distributions are computed based on the decision threshold of zero, for a channel with $z = 1$ and interference coefficients that take values from the set $\{-\delta, \delta\}$, for some positive real δ. Different scenarios are obtained by controlling the value of δ and σ. For the purpose of illustration, it is also assumed that the symbol patterns are unconstrained. Thus, the a priori probabilities for the normalized interference, appropriately shifted and scaled, follow a binomial distribution with $p = 0.5$. The changes in the behavior of the a posteriori probability distribution resulting from varying the channel and noise parameters are indicative of the shift in the error mechanism as the system transitions from one limiting case to another.

1.3.1 Large-noise Regime

The large-noise regime occurs when the noise variance σ^2 is sufficiently large relative to the variance of the ISI, σ_{ISI}^2. Then, conditioning on the signal variable Z_i, in the $Y_i = Z_i + N_i$ expression, provides little information about the received signal Y_i and the a posteriori symbol interference probabilities are approximately equal to the a

[4] i.e. the value of the signal in the event that no ISI occurs. If the transmissions are coded, z may not equal the expectation of the random variable Z_i.

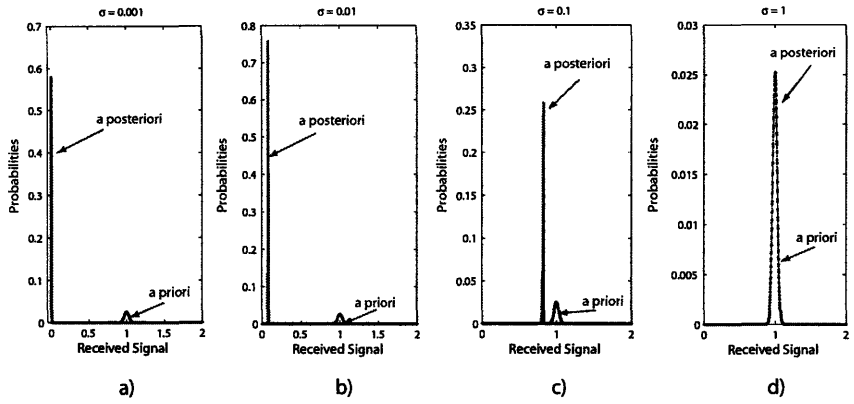

Figure 1-2: A priori and a posteriori probability distributions for a channel of length $L = 1000$ varying noise levels – a) $\sigma = 0.001$, b) $\sigma = 0.01$, c) $\sigma = 0.1$, d) $\sigma = 1$. The system in a) operates in the worst-case-dominant regime, while d) operates in the large-noise regime. The remaining cases are large-set-dominant.

priori probabilities. The large-noise regime is illustrated in Figures 1-2 d) and 1-3 e) although, depending on the context, the setup of Figure 1-3d) could be considered large-noise as well. Note that $\sigma_{ISI}^2 = 10^{-3}$ for the system of Figure 1-2 and $\sigma_{ISI}^2 = 0.1$ for that of Figure 1-3.

The relative magnitudes of noise variance and the ISI required for the system to operate in the large-noise regime also depend on the signal mean z. For a system operating "far" from the decision threshold, a tolerable amount of ISI for the large-noise regime to apply is significantly lesser than that required for a system operating closer to the decision threshold. As an illustration, consider a system with noise of variance σ^2 and let δ be the channel coefficient of smallest magnitude. If the system indeed operates in the large-noise regime, then the ratio of the a posteriori probabilities for two different ISI values will be equal to the ratio of their a priori probabilities. Now, let z be sufficiently large, so that any detection error is due to a low-probability noise event. Since cumulative probabilities in the tails of the Gaussian distribution can be approximated as

$$P(N > z) \approx \frac{e^{\frac{-z^2}{2\sigma^2}}}{z\sqrt{2\pi\sigma}} \tag{1.2}$$

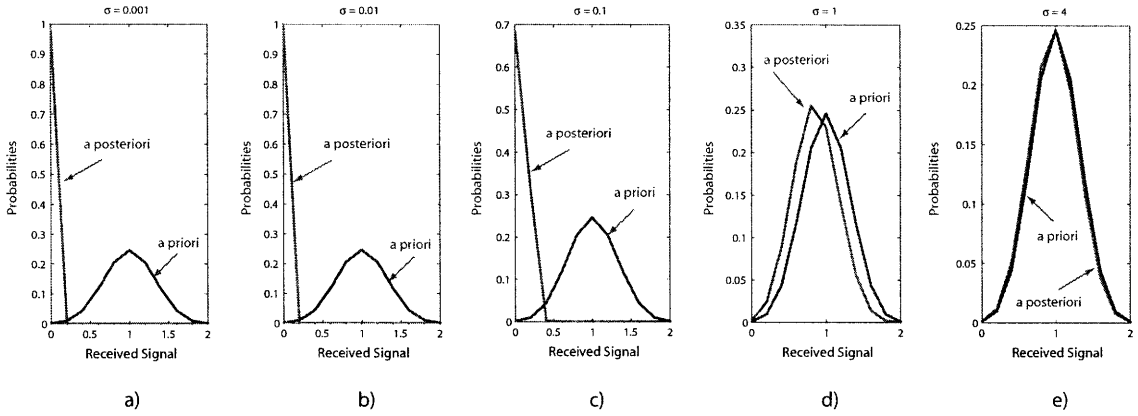

Figure 1-3: A priori and a posteriori probability distributions for a channel of length $L = 10$ varying noise levels – a) $\sigma = 0.001$, b) $\sigma = 0.01$, c) $\sigma = 0.1$, d) $\sigma = 1$, e) $\sigma = 4$. Systems a) through c) are worst-case-dominant, while e) operates in the large-noise regime. System d) is large-set-dominant, but could be considered large-noise depending on the context.

obtained by truncating[5] the corresponding asymptotic series [19], then, placing the decision threshold at zero, the ratio of the a posteriori probabilities between the signal incurring no ISI and that suffering interference of 2δ in the direction of the decision threshold is given by

$$\begin{aligned}
\frac{P(Z_i = z | Y_i < 0, X_i = 1)}{P(Z_i = z - 2\delta | Y_i < 0, X_i = 1)} &= \frac{P(Z_i = z)P(N_i > z)}{P(Z_i = z - 2\delta)P(N_i > z - 2\delta)} \\
&= \frac{P(Z_i = z)}{P(Z_i = z - 2\delta)} \times \frac{(z - 2\delta)e^{\frac{-z^2}{2\sigma^2}}}{ze^{\frac{-(z-2\delta)^2}{2\sigma^2}}} \\
&= \frac{P(Z_i = z)}{P(Z_i = z - 2\delta)} \times \frac{z - 2\delta}{z} e^{-2z\delta/\sigma^2 + 2\delta^2/\sigma^2}.
\end{aligned}$$

As z gets larger, keeping δ and σ constant, the ratio $(z - 2\delta)/z$ tends to unity, but the factor $e^{-z\delta/\sigma}$ tends to zero and the ratio of the a priori probabilities is thus *not* maintained. Thus, as z increases, the operating conditions move away from the large-noise regime.

[5]The accuracy of both the asymptotic expansion and the subsequent truncation is discussed in Section 2.2.2 of Chapter 2.

1.3.2 Worst-case-dominant Regime

The worst-case ISI occurs when a transmitted symbol pattern causes the received signal to deviate from its mean value z by the maximum possible amount Δ in the direction of the decision threshold. For a channel of length L defined by coefficients $h_0, h_1, \ldots, h_{L-1}$, where $h_1, \ldots, h_{L-1}$ cause interference, there are two possible worst-case patterns, depending on the value of the most-recently transmitted symbol X_i:

$$(X_i, \ldots, X_{i-L+1}) = (\text{sign}\,(h_0), -\text{sign}\,(h_1), -\text{sign}\,(h_2), \ldots, -\text{sign}\,(h_{L-1}))$$

and

$$(X_i, \ldots, X_{i-L+1}) = (-\text{sign}\,(h_0), \text{sign}\,(h_1), \text{sign}\,(h_2), \ldots, \text{sign}\,(h_{L-1}))$$

The system effectively operates in the worst-case-dominant regime when both of the following two conditions are satisfied:

1. The signal value affected by the worst-case ISI is at some non-negative distance away from the decision threshold, i.e. the minimum decision distance is positive.

2. The noise standard deviation, σ, is small relative to the channel coefficient of least magnitude, δ.

Then, the error events are principally due to the occurrence of the worst-case ISI coupled with a noise event. The corresponding a posteriori probability distributions for the observed interference therefore assign some large probability to the worst-case event. This occurs in Figures 1-2 a) and 1-3 a - c). Note that the a posteriori probability in Figure 1-2 b) is not concentrated on the worst-case interference, but on some adjacent interference value. Thus, whether to categorize the corresponding regime as worst-case-dominant is context-dependent.

The positive distance by which the signal mean z is separated from the decision threshold affects the allowed range for the δ/σ ratio. Applying the expression of Equation 1.2 for the cumulative probability in the tails of the Gaussian distribution, it follows that the allowed range for the values of δ/σ can be relaxed as the mean signal

value z moves away from the decision threshold. More precisely, the factor $e^{-z\delta/\sigma}$ now serves to *suppress* the a posteriori probability of symbol patterns which do not bring the signal the closest to the error region. For large z where this expression is valid, increasing z by a factor of $\alpha > 1$ allows to reduce the δ/σ ratio roughly by a factor of $1/\alpha$.

However, for some given σ, δ and z, whether a system operates in the worst-case dominant regime is also a function of the channel length L. Since several symbol patterns can cause an identical amount of ISI, both the a priori, and therefore the a posteriori, signal probability distributions take into account this multiplicity. Thus for large L, the multiplicity can bias the a posteriori distributions away from the worst-case. As an illustration, comparing the individual plots of Figures 1-2 and 1-3 yields four pairs of systems with equal σ and z, but different L. Although the δ of the plots correponding to Figures 1-3 b - c) ($L = 10$) is greater than that of the systems in Figure 1-2 b - c) ($L = 1000$), the former operate in the worst-case-dominant regime while the latter do not.

Finally, it remains to justify the non-negativity requirement on the minimum decision distance. Placing once again the decision threshold at zero, assume that $z' = z - \Delta \ < 0$ where z' represents the value of the noiseless received signal Z_i when affected by the worst-case ISI. Suppose in addition that there exists another possible value of Z_i, denoted by z'', that is also at a negative distance from the decision threshold. More precisely, there exists some possible outcome z'' such that $z' < z'' < 0$. Then, for sufficiently large L, conditioning on an error event may assign a larger a posteriori probability to z'' than to z' simply on account of its multiplicity. An example of this occuring is depicted in Figures 1-4 a) - c) below. In particular, for the system depicted in the part a) of the figure and operating in the worst-case-dominant regime, the signal mean z is reduced by 2.5δ and 4δ so that the worst-case interference crosses the decision threshold. The resulting a posteriori probability distributions, displayed in parts b) and c), are no longer worst-case-dominant, as the probability mass is centered on the interference values closer to the decision threshold.

Alternatively, a more precise argument justifying the non-negativity requirement

is available when the noise variance is sufficiently small so that the probabilities $P(N < z')$ and $P(N < z'')$ are both from the tails of the Gaussian distribution. In those conditions, the ratio of the a posteriori probabilities becomes:

$$\begin{aligned}\frac{P(Z_i = z'|Y_i < 0, X_i = 1)}{P(Z_i = z''|Y_i < 0, X_i = 1)} &= \frac{P(Z_i = z')P(N_i < -z')}{P(Z_i = z'')P(N_i < -z'')} \\ &= \frac{P(Z_i = z')P(N_i > z')}{P(Z_i = z'')P(N_i > z'')} \\ &= \frac{P(Z_i = z')}{P(Z_i = z'')} \times \frac{z''}{z'} \times \frac{1 - e^{\frac{-z'^2}{2\sigma^2}}}{1 - e^{\frac{-z''^2}{2\sigma^2}}}.\end{aligned}$$

Since $|z'| > |z''|$, the two ratios on the left-hand side are both less than unity. Thus, if z'' has an equal or greater a priori probability than z', the a posteriori probability will be biased in its favor. The regime will therefore not be worst-case-dominant.

However, note that the non-negativity requirement on the minimum distance is in principle too strict. More precisely, it is possible to envision a case where z' is the only possible negative ISI value and the next-to-worst-case ISI is sufficiently removed for the above ratio to be large and for the error expression to remain dominated by the occurrence of the worst-case ISI. While this case may be of some practical importance, a simpler definition which encompasses a large number of cases is preferable for the purpose of the subsequent development.

Quasi-worst-case-dominant Scenarios

While the previous development concerns the regime where the worst-case interference is responsible for most of the error events, such a behavior is seldom observed in practice. Instead, a more common occurrence is that of a *dichotomous channel.* The term refers to any channel of length L whose l coefficients are *more significant* than the remaining ones. The notion of significance is context-dependent. For instance, it may pertain to the confidence of the channel response measurements, or, in dispersive channels, to the fact that the first l coefficients are typically of larger magnitude. In general, the corresponding l coefficients are referred to as the *principal part of the channel,* while the remaining coefficients belong to the *secondary part of the channel.*

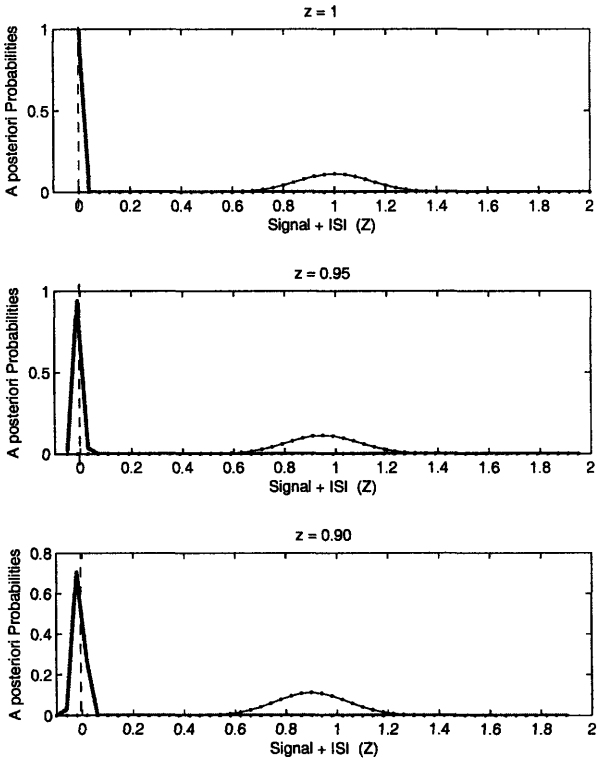

Figure 1-4: Illustrating the non-negative-minimum-distance requirement for the worst-case-dominant regime. In all three plots, $\sigma = 0.01, \delta = 0.02, L = 50$ which implies that $\Delta = 1$. The decision threshold is placed at zero as indicated by the dashed line, and the corresponding minimum decision distance is given by $z - \Delta$. The corresponding error probabilities p_{err} are included for completeness. – a)$z = 1, p_{err} = 4.5 \times 10^{-16}$. The a posteriori probability of the worst-case pattern is 0.9968, thus the regime can be considered worst-case-dominant for many practical contexts. b) $z = 0.95, p_{err} = 4.0 \times 10^{-14}$. The symbol patterns that produce second-to-worst ISI have the largest a posteriori probability. c) $z = 0.90, p_{err} = 1.5 \times 10^{-12}$. The set of symbol patterns that causes the received symbol to fall into the region $[-3\delta, 3\delta]$ dominates the a posteriori probabilities. The worst-case ISI, which causes the event $Z = -5\delta$, is not part of this set.

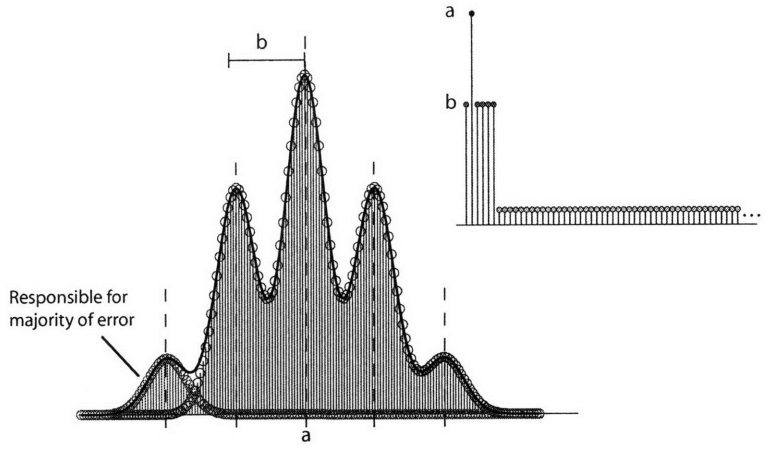

Figure 1-5: A Dichotomous Channel in the Quasi-worst-case-dominant Scenario

Note that, in general, the l coefficients need not occur consecutively, nor be restricted to a specific portion of the channel.

By considering an equivalent channel of length l, the previous characterization of the worst-case behaviors extends to worst-case interference caused by the principal part of the channel. The corresponding regime is referred to as the *quasi-worst-case-dominant regime* and is illustrated in Figure 1-5 for a two-level channel. In this example, the principal part of the channel has length $l = 5$, signal mean $z = a$ and interference coefficients of magnitude b, while the secondary part of the channel has coefficients of some lesser, unspecified magnitude. The plot illustrates only the *a priori* probability distribution of the random variable Z_i and highlights the events that aggregately dominate the a posteriori probability. The latter correspond to all the events associated with the worst-case patterns $\pm(-1,\ 1, -1, -1)$ formed by the principal part of the channel and producing signal values centered at $a - 4b$.

1.3.3 Large-set-dominant Regime

In the subsequent chapters, the large-set-dominant regime is principally considered as a default regime for all the cases that, in a given context, do not fit the two other regimes. For instance, it can be used to classify the behaviors illustrated in Figures 1-2 a - b) and, depending on the context, 1-2 c - d). It also applies to the

conditions of Figure 1-4, where the system does not operate in either the worst-case or quasi-worst-case-dominant regime due to the negative decision distance. Since the large-set-dominant scenario encompasses a range of possible a posteriori probability distributions, more general results regarding the error affecting any given symbol are difficult to formulate. Instead, Chapter 3 develops a numerical algorithm for computing probability distributions when a system cannot be considered to operate in one of the limiting cases.

However, an important general result regarding the large-set-dominant scenario can be formulated with respect to the *joint* error behaviors. Specifically, in large-set-dominant conditions that are sufficiently removed from the worst-case or quasi-worst-case scenarios, the error events on the received signals can be considered effectively independent. Since the corresponding conditions pertain to joint error statistics, rather than the a posteriori error probabilities, these cases are discussed in Section 2.3.2 of Chapter 2.

1.4 Summary

The purpose of this chapter is two-fold. First, it seeks to convey the sense of urgency which permeates many aspects of the high-speed link research, in a race to keep up with the ever-increasing demands in speed and energy-efficiency. Second, it introduces an abstracted framework suitable for a more theoretical approach to high-speed links. Modeling a high-speed link as an ISI-limited system with additive white Gaussian noise, possible error mechanisms are categorized according to three scenarios, or regimes: the large-noise, the worst-case-dominant and the large-set-dominant. This categorization plays an important part in the subsequent chapters by providing means for a significantly more rigorous analysis than that achieved in the previous work on coded high-speed links, as well as enabling the development of new, alternative error-control methods tailored for the different scenarios.

Chapter 2

Coding for High-speed Links

Most modern communication systems employ some form of coding as a technique to improve the quality of communication. These often include redundancy-based error control codes that allow for error detection or correction, run-length-limiting codes that improve the receiver's clock recovery, or DC-balancing codes that protect the timing circuitry against capacitative coupling. While inter-symbol interference (ISI) has a limited effect on the timing properties of a code, the performance of error-control codes is significantly impaired, as demonstrated in [4]. A variety of higher-complexity techniques discussed in Section 2.1.3 combat the ISI to some extent, but few are suitable for power or complexity-constrained systems.

The developments in this chapter build on the regime classification framework developed in Chapter 1 in order to further characterize the marginal and the joint error behaviors of systems with noise and inter-symbol interference. The corresponding results provide conditions under which standard error-control codes perform optimally[1] and, otherwise, lead to a new approach to error-control coding for the worst-case-dominant or quasi-worst-case-dominant regimes.

[1]That is, the conditions where the ISI has little effect on the joint error statistics and the error control suffers little or no impairement.

2.1 Preliminaries

This section overviews the basics of error-control coding and extends the previously-described high-speed link model to include coded transmissions. Previous work regarding coding for high-speed links is overviewed as well.

2.1.1 Error-control Coding

This section briefly overviews the basic principles of error-control coding of use in Chapters 2 and 3. A more thorough treatement is available in [1] or [2].

Error-control codes codes introduce controlled redundancy in order to improve the reliability of the transmission, either through forward error correction or error correction with retransmissions. Note that the corresponding stream of bits is therefore necessarily constrained. In an (n, k) binary linear block code, each n-bit codeword is obtained through some linear combination, over the binary field $\mathbb{F}_2 = \{0, 1\}$, of the underlying k information bits. In a systematic linear block code, the k information bits appear explicitly, along with the $n - k$ parity bits computed using a binary map. Linear block codes over finite fields of higher orders operate on the same principle. The most celebrated example are the Reed-Solomon codes, which are extensively used in data storage and telecommunications.

Linear block codes are intuitively simple and thus commonly provide a starting point for new applications, as further discussed in Section 2.1.3. However, the concept of error-control coding also extends to the more powerful convolutional, LDPC and turbo codes.

Regarding decoding, a system can implement hard-decision decoding or soft-decision decoding. Hard-decision decoding operates over a finite field and is thus decoupled from the detection problem. In soft-decision decoding, the real-valued signals are used in order to make more informed decisions. Although hard-decision decoding allows for relatively simple hardware implementations—for binary linear block codes, the setup is a simple threshold device folowed by delay and logic elements—the soft-decision decoding provides a performance benefit.

A notion of importance in linear block codes is that of Hamming distance. For any two codewords, the Hamming distance corresponds to the number of positions, out of n, where the codewords differ. For some coodebook C, defined as the set of allowed n-bit codewords, the minimum Hamming distance d_{H} is defined as the minimum distance between any two codewords in the codebook. Assuming hard-decision decoding and correcting to the nearest[2] codeword, a codeword will be decoded correctly if and only if there are less than $\lfloor d_{\mathrm{H}}/2 \rfloor$ detection errors in a codeword. The quantity $\lfloor d_{\mathrm{H}}/2 \rfloor$ is the error-correcting power of a code, denoted by the parameter t.

Note that in bandwidth-limited systems, an important factor of the code performance is the coding overhead. The coding overhead refers to the fact that only k bits out of the n codeword bits carry information. For a coded system operating at some signalling rate R, the equivalent information rate is thus Rk/n. When a system is severely bandwidth-limited, it can happen that an uncoded system operating at a rate of Rk/n can outperform a coded system operating at rate R. In the context of this thesis, this behavior is referred to as the *rate penalty of a code.*

2.1.2 System Model

The system model is that of the abstracted ISI-and-AWGN-limited system introduced in Chapter 1, with the addition of an encoder/decoder, as shown in Figure 2-1. Despite the fact that the depicted system implements hard-decision decoding, the developments of Chapters 2 and 3 are general, unless specified otherwise. The meaning of quantities X_i, Z_i, and Y_i is unchanged and the convolution equation, reproduced below for convenience, remains valid.

$$Z_i = \sum_{k=1}^{L} X_k h_{L-k} \tag{2.1}$$

In the above equation, X_1 is transmitted first, X_L last, and $h_0, \ldots, h_{L-1} \in \mathbb{R}$ are the channel coefficients. Note that, for the ease of notation, the communication channel is assumed to be causal, that is $h_i = 0$ for indices $i < 0$. This is also referred

[2]Again, with respect to the Hamming distance metric.

to as the channel causing no *pre-cursor ISI*. Through the remainder of the chapter, the cases where pre-cursor ISI changes the nature of the result will be discussed explicitly. Otherwise, it is to be assumed that the results hold unchanged or require trivial adjustments, such as adjustments to indexing.

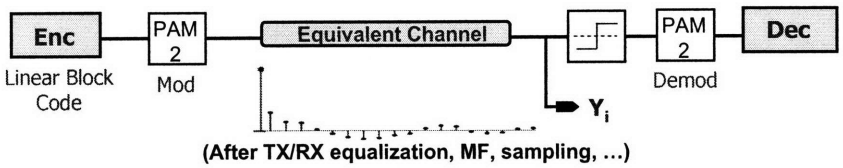

Figure 2-1: Equivalent Channel Model

Although this chapter deals with a coded high-speed link, the development focuses on the abstracted physical layer, that is, on the behavior of the system between the encoder/modulator and decoder/demodulator blocks. In this context, a *symbol* still refers to the modulated version of an individual bit, rather than the full codeword. In that sense, the effect of coding is to constrain the symbol stream, while the effect of the ISI is to introduce dependencies between the corresponding received signals. Assuming blind symbol-by-symbol MAP detection, discussed in Chapter 1, and placing the decision threshold at the origin, a detection error thus still refers to the union of the events $\{Y_i < 0 | X_i = 1\}$ and $\{Y_i > 0 | X_i = -1\}$. The notion of decoding for error-control codes, and the error rates resulting *after error correction*, is addressed through joint symbol error statistics, by considering the probability of observing more than t detection errors in a given block of n symbols.

2.1.3 Previous Work

Since coding schemes were previously considered impractical for high-speed links due to their rate penalty, most research efforts to bring more advanced communication techniques to high-speed links have focused on improved signaling/modulation techniques and equalization [5, 16]. Prior to Lee's S.M. thesis [4], two results [11][12] reported successful implementations of forward error correction or error detection codes for high-speed links. However, [4] was the first systematic study of the benefits

of such coding schemes. The principal results of [4], achieved experimentally through an FPGA[3] implementation of encoders and decoders, are the following. First, for the link channels tested, codewords with up two 10 errors occur with sufficiently high probability. Since forward error correction schemes need to provide immunity against such events, [4] concludes that the forward error correction performance gain is not high enough to justify the hardware and rate overhead. Second, burst forward error correction codes, optimized to deal with a potentially large number of errors occurring within some separation, are impractical since the typical burst length is often too large to allow for a low-overhead code. Third, assuming accurate retransmissions, all the tested error detection schemes yielded an improvement in the bit error rate of five orders of magnitude or more, including the rate penalty. Since, relative to the corresponding error correction capability, error detection requires relatively low overhead, [4] recommends the implementation of error detection codes with an automated repeat request (ARQ) scheme. The feedback path required for ARQ is available in a high-speed link through common-mode back-channel signaling [7].

The principal reason why [4],[11] and [12] rely solely on experimental results is the previous lack of a suitable analytical framework as well as a lack of alternative performance evaluation techniques for coded high-speed links. While it was previously recognized that the inter-symbol interference is an important error mechanism in high-speed links [6], no previous analytical characterization of different couplings between the noise and the interference, and the resulting effect on the nature and frequency of error occurrences, is available. The previous work in performance evaluation of high-speed links is discussed in Section 3.2 of the subsequent chapter.

The general subject of communicating in ISI-dominated environments has been addressed in several different contexts. The standard approach consists of decoupling the equalization and coding. More precisely, drawing from variety of equalization techniques [3], the problem becomes that of designing an optimal equalizer to minimize the ISI and designing optimal codes for ISI-free operation. Although this has been known to combat the bandwidth limitations to a practical degree, the tech-

[3]Field Programmable Gate Array.

nique suffers from the rate penalty whose effects vary with the severity of the ISI. A widely-acknowledged class of channels with severe ISI are encountered in magnetic recording. Based on the differentiation step inherent in the read-back process, magnetic recording systems are modeled as partial response channels [20]. The partial response channels of interest in magnetic recording are integer or binary-valued, first studied in [47]. Since for real-valued channels with noise and ISI, operating in the worst-case or quasi-worst-case-dominant regime reduces the communication channel to a binary signature, the results pertaining to binary-valued partial response channels are of interest for the present development. An example of such a channel is the Extended Partial Response 4 (E^2PR4) channel.

A variety of communication techniques has been developed to improve the communication over partial response channels and other ISI-limited environments. These include decision-feedback-based techniques with coset codes [24, 25], Tomlinson-Harashima precoding [27, 28], vector coding [29], partial response maximum likelihood [21–23], and numerous extensions of coding concepts to partial response channels, where the most recent ones include [30–36] among others. However, the most relevant link to the pattern-eliminating codes introduced in this chapter is the work on distance-enhancing constraint codes for partial response channels.

The work on distance-enhancing constraint[4] codes spurred from an observation [37] that a rate 2/3 $(d, k) = (1, 7)$ run-length-limiting (RLL) code provides a coding gain of 2.2 dB on the E^2PR4 channel, where the coding gain is measured with respect to the squared Euclidean distance. Shortly after, maximum-transition-run (MTR) codes were introduced in [38] and demonstrated to yield potentially large coding gains. Similarly, the last twelve years have witnessed a wealth of development in *distance-enhacing constraint-codes* for the partial response channels. Some of the principal results are thoroughly reviewed in [39], while the more recent contributions include [40–45] among others.

Much like the pattern-eliminating codes, the distance-enhancing constraint codes

[4]For convenience, the term *constraint coding* refers to any code whose sole purpose is not error correction. Namely, such codes include RLL and MTR codes.

yield a coding gain by preventing the occurrence of harmful symbol patterns. However, several fundamental differences distinguish the two types of codes. Structure-wise, the pattern-eliminating codes are systematic, while the distance-enhancing constraint codes are not. Thus, both the proof techniques and the results differ between the two cases. More importantly, the pattern-eliminating codes are optimized to provide an improvement to the minimum decision distance, while this criterion is secondary in the distance-enhancing constraint codes. Note that the distance-enhancing benefit of constraint codes has only been reported for E^2PR4 and E^3PR4 channels, as the partial response channels of higher-orders are not binary. However, the codes may yield a benefit over a wider range of channel signatures[5], a topic which remains unexplored since binary channels of arbitrary signatures are not encountered in magnetic recording. Since distance-enhancing constraint codes are primarily designed for timing purposes, it is unlikely that their distance-enhancing potential rivals that of pattern-eliminating codes. However, a more precise comparison remains to be performed. Finally, due to their non-systematic nature and non-trivial decoding, the distance-enhancing constraint codes may require more hardware than the pattern-eliminating codes but a more precise characterization is needed.

Finally, note that in the worst-case-dominant or quasi-worst-case-dominant regime, the problem reduces to that of dealing with a discrete noiseless channel (DNC), first studied by Shannon [13]. The DNC is characterized by a set of allowed transmitted symbol sequences, or alternatively by their complement, that is, the set of forbiden sequences. Thus, in a sense, error-free operation is achieved on the DNC by preventing the occurrence of symbol patterns from some given set. Shannon showed that the capacity C of the DNC is given by

$$C = \lim_{t \to \infty} \frac{\log n(t)}{t}$$

where $n(t)$ denotes the number of allowed sequences of length t. Although not as ubiqutous as the noisy channel models, the DNC continues to generate interest al-

[5] A channel signature of a real-valued channel response is the underlying binary channel. Channel signatures are first defined in Section 2.3.

most sixty years after the publication of [13]. For instance, [14] revisits, clarifies and further generalizes the most important theorems on the DNC, while [15] considers a combinatorial approach to computing the capacity of the DNC. However, the most important body of work on the DNC relates to the development of constraint codes. Specifically, Shannon's results apply directly to most forms of constraint coding over ISI-limited channels and have therefore found extensive application in magnetic recording channels, where constraint codes are commonly used.

2.2 Coding on ISI-and-AWGN-limited Systems — Symbol Error Probabilities

This section develops general results regarding the effect of constraining the transmit alphabet on the error probabilities of individual symbols. It is important to note that the error events considered are those occurring on the symbol value, that is *prior to decoding.* Analyzing the effect of a code on the systems performance *after decoding* requires information about the joint error behaviors. The latter are the subject of Section 2.3. Note that the present development deals with arbitrary constraints on the transmitted sequence $X_i, X_{i-1}, \ldots$, and thus applies to any code, including the linear block codes and the constraint codes.

It is convenient to first consider an uncoded system and then observe what happens with the addition of a code. Following the notation of Chapter 1, consider the uncoded system characterized by zero-mean additive white Gaussian noise of variance σ^2 and a channel of some finite length L. Observing the transmitted symbol X_i, suppose that a symbol error occurs with probability p_{err}. Since the transmitted symbols are assumed to occur independently, with values drawn from the set $\{-1, 1\}$ with equal probability, since the noise affecting the signal is independent of the signal and since it has zero mean, placing the decision threshold at the origin causes the symbol detection error

p_{err} to be independent of the value of the transmitted symbol. That is,

$$\begin{aligned} p_{err} &= \mathrm{P}(Y_i < 0 | X_i = 1)\mathrm{P}(X_i = 1) + \mathrm{P}(Y_i > 0 | X_i = -1)\mathrm{P}(X_i = -1) \quad (2.2a) \\ &= \mathrm{P}(Y_i < 0 | X_i = 1) \quad (2.2b) \end{aligned}$$

for any $i \in \mathbb{Z}$. In addition, since the noise is white and the transmitted symbols unconstrained, the errors occuring on distinct symbols are identically distributed[6]. Thus, to simplify the exposition, all the results of this section concern the transmitted symbol at some time index $i \in \mathbb{Z}$ and are conditioned on $X_i = 1$. Then, let $\mathbf{X} = (X_{i-1}, \ldots, X_{i-L+1})$ be the string of $L-1$ previously[7] transmitted symbols.

Employing the inner product notation as a shorthand, let $\mathbf{h} = (h_1, \ldots, h_{L-1})$ and rewrite Equation 2.1 as

$$Y_i = h_0 X_i + \langle \mathbf{X}, \mathbf{h} \rangle + N_i.$$

The following expressions for the error probability are of some use in the upcoming development.

$$\begin{aligned} p_{err} &= \sum_{\mathbf{x} \in \{-1,1\}^{L-1}} \mathrm{P}(Y_i < 0 | \mathbf{X} = \mathbf{x}, X_i = 1)\mathrm{P}(\mathbf{X} = \mathbf{x} | X_i = 1) \quad (2.3a) \\ &= \sum_{\mathbf{x} \in \{-1,1\}^{L-1}} \mathrm{P}\left(N_i < -\left(h_0 + \langle \mathbf{x}, \mathbf{h} \rangle\right)\right) \mathrm{P}(\mathbf{X} = \mathbf{x} | X_i = 1) \quad (2.3b) \end{aligned}$$

where, as previously defined, $N_i \sim \mathrm{N}(0, \sigma^2)$ represents the noise on the received signal. In addition, note that for an uncoded system,

$$\mathrm{P}(\mathbf{X} = \mathbf{x} | X_i = 1) = \mathrm{P}(\mathbf{X} = \mathbf{x} | X_i = -1) = 2^{-L+1} \quad \forall\, \mathbf{x} \in \{-1, 1\}^{L-1}$$

At this point, it is convenient to distinguish between the large-noise, worst-case-dominant and large-set-dominant scenarios. The development of all three sections is general, that is, does not rely on any a priori classification. To the contrary, it

[6]But are in general not independent unless separated by more than L symbols.

[7]Note that this notation differs from that of Section 2.3 where $\mathbf{X} = (X_i, X_{i-1}, \ldots, X_{i-L+1})$, that is, where vector $\mathbf{X}$ includes the most-recently transmitted symbol. The choice of excluding X_i simplifies the notation when conditioning on the event $X_i = 1$.

develops practical results that can be used to classify a system according to the three interference-and-noise scenarios in the context of its symbol error probability. The worst-case-dominant scenario is addressed first, since it follows most immediately from the previous development.

2.2.1 Symbol Error Probabilities in the Worst-case-dominant Regime

Conditioned on $X_i = 1$, let $\mathbf{x}_{wc} \in \{-1, 1\}^{L-1}$ denote the symbol pattern that minimizes the distance to the decision threshold of the recieved signal in the absence of noise, that is,

$$\langle \mathbf{x}_{wc}, \mathbf{h} \rangle = \min_{\mathbf{z} \in \{-1,1\}^{L-1}} \langle \mathbf{z}, \mathbf{h} \rangle$$

Assuming for the time being that the channel has no coefficients of zero magnitude and that, without loss of generality, $h_0 = 1$, it follows that

$$\mathbf{x}_{wc} = (-\operatorname{sign} h_1, -\operatorname{sign} h_2, \ldots, -\operatorname{sign} h_{L-1}). \tag{2.4}$$

Expanding Equation 2.3a,

$$\begin{aligned} p_{err} &= P(Y_i < 0 | \mathbf{X} = \mathbf{x}_{wc}, X_i = 1) \mathrm{P}(\mathbf{X} = \mathbf{x} | X_i = 1) \\ &\quad + \sum_{\mathbf{x} \in \{-1,1\}^{L-1},\, \mathbf{x} \neq \mathbf{x}_{wc}} \mathrm{P}(Y_i < 0 | \mathbf{X} = \mathbf{x}, X_i = 1) \mathrm{P}(\mathbf{X} = \mathbf{x} | X_i = 1). \end{aligned} \tag{2.5}$$

Let $f : 0 < f \leq 1$ denote the a posteriori probability, conditioned on $X_i = 1$ and on the event $Y_i < 0$, of the $L - 1$ previously transmitted symbols forming the worst-case pattern $\mathbf{x}_{wc}$. More precisely,

$$\mathrm{P}(\mathbf{X} = \mathbf{x}_{wc} | Y_i < 0, X_i = 1) = f \tag{2.6}$$

The *pure* worst-case-dominant regime thus happens in the limit $f \to 1$. However, the following development makes no assumption on the value of f. Instead, the concluding

part of this section offers practical guidelines on when f is sufficiently large for the error to be attributed, with sufficient confidence, uniquely to the occurrence of the worst-case ISI.

Now, Equation 2.2b and a straightforward reformulation of the above conditional probabilities yields

$$p_{err} = \frac{\mathrm{P}(\{\mathbf{X} = \mathbf{x}_{wc} \bigcap Y_i < 0\}|X_i = 1)}{f} \tag{2.7a}$$

$$= \frac{\mathrm{P}(Y_i < 0|\mathbf{X} = \mathbf{x}_{wc}, X_i = 1)\mathrm{P}(\mathbf{X} = \mathbf{x}_{wc}|X_i = 1)}{f} \tag{2.7b}$$

$$= \frac{\mathrm{P}(Y_i < 0|\mathbf{X} = \mathbf{x}_{wc}, X_i = 1)2^{-L+1}}{f} \tag{2.7c}$$

Comparing the above equation to Equation 2.5, it follows that the portion of error *not due* to the worst-case ISI is given by

$$\sum_{\mathbf{x}\in\{-1,1\}^{L-1},\, \mathbf{x}\neq\mathbf{x}_{wc}} \mathrm{P}(Y_i < 0|\mathbf{X} = \mathbf{x}, X_i = 1)2^{-L+1} = (1-f)p_{err} \tag{2.8}$$

Finally, it is also convenient to express f as

$$f = \frac{P(Y_i < 0|\mathbf{X} = \mathbf{x}_{wc}, X_i = 1)}{P(Y_i < 0|\mathbf{X} = \mathbf{x}_{wc}, X_i = 1) + \sum_{\mathbf{x}\in\{-1,1\}^{L-1}, \mathbf{x}\neq\mathbf{x}_{wc}} \mathrm{P}(Y_i < 0|\mathbf{X} = \mathbf{x}, X_i = 1)} \tag{2.9}$$

Now consider the effect of adding a code to the previously-described system, where the code need not be a block code nor be linear. For a given transmitted symbol X_i, the code constraints imposed on symbol sequences reduce the set of allowed symbol history vectors $\mathbf{X} = (X_{i-1}, \ldots, X_{i-L+1})$ according to the value of X_i and its position in the codeword. Thus, in general

$$\mathrm{P}(\mathbf{X} = \mathbf{x}|X_i = 1) \neq \mathrm{P}(\mathbf{X} = \mathbf{x}|X_i = -1)$$

$$\text{and } \mathrm{P}(\mathbf{X} = \mathbf{x}|X_i = 1) \neq \mathrm{P}(\mathbf{X} = \mathbf{x}|X_j = 1), \text{ when } i \neq j.$$

Note that to reduce unnecessary redundancy, the subsequent developments only consider unilateral quantities, conditioned on $X_i = 1$. The analogous quantities for the

case where $X_i = -1$ are obtained in a similar manner.

Referring to the transmitted symbol $X_i = 1$ for some $i \in \mathbb{Z}$, adding a code to the system has two possible outcomes: by constraining the set of allowed transmitted patterns, the code either prohibits the event $(X_{i-1}, \ldots, X_{i-L+1}) = \mathbf{x}_{wc}$ or it does not. The following sections analyze each case individually.

Coded Performance with the Uncoded Worst Case Present

For a given code, conditioned on the transmitted symbol $X_i = 1$, let Ψ_i denote the set of allowed transmitted symbol sequences $\mathbf{X} = (X_{i-1}, \ldots, X_{i-L+1})$. The present development concerns the case where $\mathbf{x}_{wc} \in \Psi_i$ and $\mathbf{x}_{wc}$ is the uncoded worst-case symbol pattern defined previously. The analogue to the a posteriori probability f is $f_i' : \ 0 < f_i' \leq 1$ given by

$$\mathrm{P}(\mathbf{X} = \mathbf{x}_{wc} | Y_i < 0, X_i = 1) \ = \ f_i'. \tag{2.10}$$

where all probabilities from now on refer the constrained sample space. Also consider the unilateral error probability

$$p_{err}' = \mathrm{P}(Y_i < 0 | X_i = 1).$$

Noting that

$$\mathrm{P}(\mathbf{X} = \mathbf{x} | X_i = 1) = |\Psi_i|^{-1} \tag{2.11}$$

where the $|\Psi_i|$ denotes the cardinality of the set Ψ_i, and rewriting Equations 2.3b and 2.7c accordingly yields

$$p_{err}' \ = \ \sum_{\mathbf{x} \in \Psi_i} \mathrm{P}(N_i > (h_0 + \langle \mathbf{x}, \mathbf{h} \rangle)) \, |\Psi_i|^{-1} \tag{2.12}$$

$$= \ \frac{\mathrm{P}(Y_i < 0 | \mathbf{X} = \mathbf{x}_{wc}, X_i = 1) \, |\Psi_i|^{-1}}{f_i'}. \tag{2.13}$$

Similarly, Equation 2.9 can be expressed as

$$f_i' = \frac{P(Y_i < 0|\mathbf{X} = \mathbf{x}_{wc}, X_i = 1)}{P(Y_i < 0|\mathbf{X} = \mathbf{x}_{wc}, X_i = 1) + \sum_{\mathbf{x}\in\Psi_i, \mathbf{x}\neq\mathbf{x}_{wc}} \mathrm{P}(Y_i < 0|\mathbf{X} = \mathbf{x}, X_i = 1)} \tag{2.14}$$

The above expression leads to the following useful result.

Theorem 1. *For some uncoded system, let $f : 0 < f \leq 1$ denote the a posteriori probability of the worse case symbol pattern $\mathbf{x}_{wc}$, conditioned on $X_i = 1$. After imposing some set of constraints on the transmitted symbols, assume that $\mathbf{x}_{wc} \in \Psi_i$ for some $i \in \mathbb{Z}$. Letting $f_i' : 0 < f' \leq 1$ be the corresponding a posteriori probability of the same worst-case symbol pattern $\mathbf{x}_{wc}$, it follows that*

$$f_i' \geq f \tag{2.15}$$

Proof. Compare the above Equation 2.14 with Equation 2.9. Direct comparison is allowed because the probabilities in both expressions refer only to the sample space of the random variable N_i. In other words, by conditioning the received signal on some symbol pattern, the only remaining uncertainty is due to noise. Since the latter is assumed to be independent of the signal, its sample space is not affected by any code constraints. Now notice that the two equations have the same numerator but that

$$\sum_{\mathbf{x}\in\{-1,1\}^{L-1}, \mathbf{x}\neq\mathbf{x}_{wc}} \mathrm{P}(Y_i < 0|\mathbf{X} = \mathbf{x}, X_i = 1) \geq \sum_{\mathbf{x}\in\Psi_i, \mathbf{x}\neq\mathbf{x}_{wc}} \mathrm{P}(Y_i < 0|\mathbf{X} = \mathbf{x}, X_i = 1)$$

since restricting the set of allowed symbol patterns only removes elements from the sum. The result follows. □

Since the above theorem holds for any $0 < f \leq 1$, specializing it to the case where, given a context, f is sufficiently large for the regime to be considered worst-case dominant, leads to the following result.

Corollary 1. *Given an uncoded system operating in the worst-case-dominant regime, observe the symbol $X_i = x \in \{-1, 1\}$ for some $i = 1, \ldots, n$. If imposing code constraints on the system does not prohibit the occurrence of the worst-case pattern preceding symbol X_i, then the error on X_i is still dominated by the worst-case interference event.*

Reverting to the general case of $0 < f \leq 1$ and seeking to express the new symbol error probability with that of an uncoded system yields the following result.

Theorem 2. *Under the conditions of Theorem 1, let p_{err} and p'_{err} denote unilateral symbol error probabilities of the uncoded and coded systems respectively. Then,*

$$\frac{p'_{err}}{p_{err}} = \frac{2^{L-1}}{|\Psi_i|}\frac{f}{f'_i} \tag{2.16}$$

Thus,

$$f\frac{2^{L-1}}{|\Psi_i|} \leq \frac{p'_{err}}{p_{err}} \leq \frac{2^{L-1}}{|\Psi_i|} \tag{2.17}$$

Proof. The equality follows from Equations 2.7c and 2.13. The upper inequality is a consequence of Theorem 1, while the lower follows letting $f'_i = 1$. □

It follows that, in the limit of $f \to 1$, the coded symbol error probability p' equals the uncoded probability p, adjusted by a scaling factor that reflects the increase in the relative weight of each remaining symbol pattern.

Coded Performance with the Uncoded Worst Case Removed

Now consider the case where, for some index $j \in \mathbb{Z}$, the worst-case pattern $\mathbf{x}_{wc}$ is prohibited from preceding $X_j = 1$. In other words, $\mathbf{x}_{wc} \notin \Psi_j$. The new unilateral error probability p''_{err} is given by

$$p''_{err} = \mathrm{P}(Y_j < 0 | \mathbf{X} = \mathbf{x}, X_j = 1) \tag{2.18}$$

$$= \sum_{\mathbf{x} \in \Psi_j} \mathrm{P}(Y_j < 0 | \mathbf{X} = \mathbf{x}, X_j = 1)\mathrm{P}(\mathbf{X} = \mathbf{x} | X_j = 1) \tag{2.19}$$

where

$$\mathrm{P}(\mathbf{X} = \mathbf{x}|X_j = 1) = |\Psi_j|^{-1} \tag{2.20}$$

and $|\Psi_j|$ denotes the cardinality of the set Ψ_j.

The following is the counterpart to Theorem 2 of the previous section.

Theorem 3. *For some uncoded system, let $f : 0 < f \leq 1$ denote the a posteriori probability of the worse case symbol pattern $\mathbf{x}_{wc}$, conditioned on $X_j = 1$. After imposing some set of constraints on the transmitted symbols, assume that $\mathbf{x}_{wc} \notin \Psi_j$ for some $j \in \mathbb{Z}$. Then for the new unilateral symbol error probability, p''_{err} given by Equation 2.18, it holds that*

$$\frac{p''_{err}}{p_{err}} \leq \frac{2^{L-1}}{|\Psi_j|}(1 - f)$$

Proof. Since

$$\sum_{\mathbf{x} \in \Psi_j} \mathrm{P}(Y_j < 0|\mathbf{X} = \mathbf{x}, X_j = 1) \quad \leq \sum_{\mathbf{x} \in \{-1,1\}^{L-1},\, \mathbf{x} \neq \mathbf{x}_{wc}} \mathrm{P}(Y_j < 0|\mathbf{X} = \mathbf{x}, X_j = 1)$$

the result follows by applying Equation 2.8 and Equation 2.19. □

Note that the above bound may not be of practical use for moderately small values of f as the ratio $\leq \frac{2^{L-1}}{|\Psi_i|}$ is potentially large for high-overhead codes. However, as $f \to 1$, the bound becomes tight. As further discussed in Section 2.2.1, the corresponding result is of significant practical importance as it provides the foundation for a new approach to coding for worst-case-dominant scenarios.

Extensions to Quasi-worst-case-dominant Scenarios

The previous results extend to the case of a dichotomous channel introduced in Chapter 1. Note that, as previously discussed, the dichotomous nature of the channel is purely a practical concept—it conveys how much of the channel response is *significant* in some context.

By redefining the meaning of symbols f and L, many of the previously developed results extend immediately to the case where the significant coefficients are consecutive. Specifically, consider a channel of length L where the corresponding coefficients can be separated into two parts: the principal part of length L' and the secondary part of length $L - L'$. By redefining the worst-case pattern in terms of the principal part alone and taking into account the fact that the worst-case pattern now has multiplicity $2^{L-L'}$, the resulting expressions for the uncoded system hold unchanged. In particular, note that , when $f = 1$, or in the limit when $f \rightarrow 1$ if the noise has infinite support, the dichotomous channel is said to operate in a quasi-worst-case-dominant scenario.

For the coded performance, restricting the set of allowed symbol patterns Ψ_i to patterns formed by the principal part of the channel alone is not sufficient to generalize all of the corresponding expressions. In particular, different patterns formed by the principal part of the channel can have different multiplicities due to the effect of the code constraints on the secondary part of the channel. However, this matters little from the practical perspective since the coefficients in the secondary part of the channel are typically considered negligible.

Practical Implications

This section revisits the previous expressions and derives practical results regarding the performance of error-correcting codes for high-speed links. Proceding in the logical order, consider the question of how large f needs to be for the error mechanism to be considered worst-case dominant in the context of symbol error probabilities. Note that the following discussion applies to any system, coded or uncoded. Moreover, note that the worst-case pattern need not be that of Equation 2.4. That is, when the former is prohibited from occuring, the new worst-case pattern becomes what was previously the second-to-worst symbol pattern, and so on.

Suppose that $f < 1$ and denote by p_{err} the true error probability. Furthermore, denote by $\tilde{p}_{err}$ the approximate error probability computed under the assumption

$f = 1$, that is

$$\tilde{p}_{err} = \mathrm{P}(Y_i < 0 | \mathbf{X} = \mathbf{x}_{wc}, X_i = 1)\mathrm{P}(\mathbf{X} = \mathbf{x}_{wc} | X_i = 1).$$

The error inherent to the approximation is then given by

$$\frac{p_{err}}{\tilde{p}_{err}} = \frac{1}{f}$$

where the above expression follows from Equation 2.7b. For instance, when $f \geq 0.1$, the approximation differs by at most an order of magnitude. This type of accuracy is adequate in high-speed links, since a system operating at error probabilities of 10^{-15}, for instance, has roughly the same performance as that operating at 10^{-14}. Also note that, since $0 < f \leq 1$,

$$\tilde{p}_{err} \leq p_{err}.$$

Thus, the $f = 1$ approximation underrepresents the true error probability by a factor of $1/f$.

The above result, in conjuction with those of the previous sections, is of particular use in performance estimation of coded systems at low error rates. For an uncoded system limited only by noise and inter-symbol interference, computing accurate error probabilities is straightforward even in the tails of the error distributions, while the same quickly becomes prohibitive for coded systems when potentially long codewords and channel responses are to be taken into account. Thus, while it's difficult to accurately compute p'_{err}, accurate results for f and p_{err} are readily available. Furthermore, note that the quantity, $|\Psi_i|$ for a coded system is entirely determined by the nature of the code and the length of the channel response; for instance, for a binary linear block code, $|\Psi_i| = 2^l$ where l is the number of non-parity symbols out of the L symbols transmitted at time indices $i, i-1, \ldots, i-L+1$.

First consider the case where the code *does not prevent* the worst-case symbol pattern from preceding symbol X_i for some time index i, that is, $\mathbf{x}_{wc} \in \Psi_i$. Then, given an accurately computed value of f, it is convenient to use the inequalities of

Theorem 2, reproduced below for convenience, to bound p'_{err}.

$$f\frac{2^{L-1}}{|\Psi_i|} \leq \frac{p'_{err}}{p_{err}} \leq \frac{2^{L-1}}{|\Psi_i|}$$

As $f \to 1$, the bound becomes tight. More practically, the bound can be used as an approximation for sufficiently large f. For instance, when $f = 0.1$, the approximation $p_{err} \approx \frac{2^{L-1}}{|\Psi_i|}$ may overestimate the new error probability by at most an order of magnitude. Also note that in the case when the code does not prevent the worst-case symbol pattern from preceding symbol X_i, Theorem 1 states that, if f is large enough for the error in the uncoded system to be considered worst-case-dominant with sufficient accuracy, then the error occuring on symbol X_i in the coded system is also worst-case-dominant.

Now consider the case where the code constraints *do prevent* the worst-case symbol pattern from preceding symbol X_i, that is, $\mathbf{x}_{wc} \notin \Psi_j$. The corresponding bound, formulated in Theorem 3 is reproduced below for convenience.

$$\frac{p''_{err}}{p_{err}} \leq \frac{2^{L-1}}{|\Psi_i|}(1-f)$$

Note that the above bound may become too pessimistic for codes with large overhead. However, the bound becomes tight as $f \to 1$, when, as expected, $p''_{err} \to 0$. This result provides the foundation of the pattern-eliminating codes introduced in Section 2.4. Since the behavior of interest concerns the conditions where f is sufficiently close to unity, developing a tighter bound at lower f, although possible, is in a sense superfluous. Furthermore, note that, regardless of the value of f, the quantity p''_{err} can be accurately computed using the numerical algorithm introduced in Chapter 3.

Finally, note that, for an (n, k) block code for instance, it is possible to have both $\mathbf{x}_{wc} \in \Psi_i$ and $\mathbf{x}_{wc} \notin \Psi_j$ for some $1 \leq i, j \leq n,\ i \neq j$. In fact, Section 2.4.2 introduces new (n, k) block codes whose sole purpose is ensuring that $\mathbf{x}_{wc} \notin \Psi_j$ *for all* $1 \leq j \leq k$. The question of how large f needs to be for such a code to provide some significant benefit is also addressed in that section.

2.2.2 Symbol Error Probabilities in the Large-noise Regime

Since the large-noise scenario treats inter-symbol interference as negligible, the corresponding symbol error probability for both coded and uncoded systems is trivially given by

$$p_{err} = \mathrm{P}(N_i < -h_0),$$

where the notation follows that of the preceding sections. The present development thus focuses on practical guidelines regarding when a system can be considered to operate in this scenario. For an uncoded system with ISI, the error inherent to assuming a large-noise scenario, expressed as a ratio of resulting error probability to the true error probability, is given by

$$\text{approximation error } = \frac{\mathrm{P}(N_i < -h_0)}{\sum_{\mathbf{x}\in\{-1,1\}^{L-1}} \mathrm{P}(N_i < -h_0 - \langle \mathbf{x}, \mathbf{h}\rangle)\ \mathrm{P}(\mathbf{X} = \mathbf{x} | X_i = 1)}$$

where $h_0 > 0$ by an earlier assumption and Equation 2.3b is used to compute the true error probability. A similar expression holds in the coded case, with the exception that the resulting probability is unilateral and that the symbol patterns prohibited by the code are not considered in the sum. However, the following simpler criterion may be more convenient, at the expense of being more restrictive. For a system with weak interference, an error event must be due to the tails of the noise distribution if the system is to operate at a low error rate[8]. Let Δ denote some *likely*[9] maximum signal deviation due to the ISI in the absence of noise. Note that the magnitude of Δ depends on the variance of the ISI. For instance, for a channel with interference coefficients on the order of $1/L$, where L is the length of the channel, the variance of the ISI equals $\frac{L-1}{L^2}$. Thus, for a relatively long channel with relatively weak interference coefficients, the quantity $h_0 - \Delta$ is typically much larger than the maximum signal deviation due to ISI, given in this example by $h_0 - \frac{L-1}{L}$.

Now let p_{err} be the interference-free error probability, that is, $p_{err} = \mathrm{P}(N_i < -h_0)$, and let $\tilde{p}_{err}$ be an exaggeration of the true error probability, that is, $\tilde{p}_{err} = \mathrm{P}(N_i <$

[8] As a reminder, high-speed links operate at error probabilities on the order of 10^{-15}

[9] i.e. the probability that $|Z| > \Delta$ is sufficiently small to be negligible in a given context.

$-h_0 + \Delta$). Then, applying the approximation to the tail of the normal CDF, given by Equation 1.2 of Chapter 1, it follows that

$$\frac{p_{err}}{\tilde{p}_{err}} \approx e^{-\Delta(h_0-\Delta/2)/\sigma^2} \frac{h_0+\Delta}{h_0}.$$

Note that the above ratio tends to unity as $\Delta \to 0$. For h_0 and $h_0 - \Delta$ adequately large so that the approximations are sufficiently accurate and assuming, in addition, that $\Delta \leq h_0/2$, it follows that

$$e^{-\Delta h_0/\sigma^2} \leq \frac{p_{err}}{\tilde{p}_{err}} \leq \frac{3}{2} e^{-\frac{3}{4}\Delta h_0/\sigma^2}.$$

The above expression is useful in that it provides a symbolic range of adequate values of Δ, given some desired accuracy, without relying on the precise value of h_0. For instance, for sufficiently large h_0 so that the above expression applies, letting $\Delta = \frac{\sigma^2}{10h_0}$ yields that $0.9 \leq \frac{p_{err}}{\tilde{p}_{err}} \leq 1.4$.

A final comment on the accuracy of the approximation to the normal CDF, used in the above expression, is in order. Equation 1.2 is obtained by truncating an asymptotic series. There are thus two issues involved: the truncation of a power series and, more difficult, the accuracy of an asymptotic expansion in the region of interest. While relevant results are likely available in the literature, the following numerical example is offered as an illustration. The accuracy of the truncated asymptotic approximation for the normal cumulative distribution function at 5σ away from the mean in the negative direction, corresponding to a cumulative probability of 10^{-7}, is approximately 4%. Furthermore, numerical simulations[10] suggest that the error decreases monotonically as the argument moves further into the tail of the distribution. Thus the approximation is adequate for the error ranges of interest.

[10]The error is evaluated in the interval $[\sigma,8\sigma]$ in increments of 10^{-3}.

2.2.3 Symbol Error Probabilities in the Large-set-dominant Regime

After excluding the large-noise, worst-case-dominant and quasi-worst-case dominant cases from the pool of possible interference-and-noise scenarios, the remainder is a set of possible situations where the error expression is dominated by some large set of events. Although a variety of asymptotic results can be derived in this case, the accuracy of the resulting approximations is more difficult to control than in the previous scenarios. Instead, Chapter 3 offers a practical numerical method of accurately evaluating the symbol error probabilities in these and other scenarios.

2.3 Coding on ISI-and-AWGN-limited Systems — Joint Error Behavior

In order to evaluate the benefits of coding for channels with inter-symbol interference it is not sufficient to consider only the marginal symbol error probabilities. The present section thus focuses on characterizing joint error behaviors. Shifting the interest from the marginal behaviors of individual symbols to the joint behavior of n consecutive symbols has the effect of also transferring the attention away from the code constraints and towards the characteristics of the channel response. The conclusions still differ depending on whether the system operates in the large-noise scenario, the worst-case-dominant scenario or the large-set scenario.

In the large-noise scenario, errors on different symbols are considered to occur independently regardless of the channel response or the code constraints, as long as the interference is sufficiently small compared to the noise variance and the noise spectrum has sufficiently large bandwidth. The following sections thus focus on the remaining two scenarios.

The present development concerns an (n, k) block code, not necessarily systematic nor linear, where the value of k is irrelevant. In order to reduce the notational burden of dealing with two sets of indices, denoting the location of the symbol in time and

within a given codeword, it is convenient to tolerate a slight abuse of notation. The difficulty arises from the fact that, for an (n, k) code, the first context allows for any $i \in \mathbb{Z}$, while the second limits the range of the index to $i \in \{1, 2, \ldots, n\}$. The solution is thus to align the time axis with the location of the first symbol in a codeword and let the indices of the previously transmitted symbols denote time indices uniquely.

2.3.1 Joint Error Behavior in the Worst-case-dominant Regime

Consider the case when the symbol error expression is dominated by the occurrence of the worst-case pattern. Following the notation of Section 2.2.1, let f denote the a posteriori probability, conditioned on an error event, of a worst-case symbol pattern. Then, in the limit of $f \to 1$, the *channel's signature*, that is the equivalent binary channel obtained by taking the sign of the original channel's coefficients, *entirely determines the joint error statistics.* However, showing this formally requires some subtlety. From the definition of f (Equation 2.6) , it is obvious that

$$\lim_{f \to 1} \mathrm{P}(\mathbf{X} = \mathbf{x}_{wc} | Y_i < 0, X_i = 1) = 1.$$

Yet, the above result does not imply that, in the limit, the probability of errors due to non-worst-case interference events is zero. By considering the limit of small noise rather than large f, it may be possible to demonstrate the stronger result. However, the intricacies of demonstrating almost sure convergence conditioned on an event of vanishing probability would distract from the mission of this thesis, written with the practicioner in mind. In practice, the worst-case-dominant scenario simply implies that the errors due to non-worst-case interference events are sufficiently unlikely to be considered negligible.

For the sake of rigor, however, it is convenient to visualize the following scenario. Suppose that the worst-case symbol pattern places the received signal at some positive distance d away from the decision threshold and the next possible interference value is at some positive distance $d + \delta$ away from the decision threshold. Assuming that the noise has some finite support which extends in the negative direction to some

value $-d'$, letting $-d-\delta < d' \leq -d$ yields a case where an error can occur only for worst-case interference. It is convenient to employ the shorthand $f = 1$ to refer to this setup. This visualization has the benefit of allowing the following development to focus on properties of worst-case patterns without obscuring it with extraneous probabilistic overtones.

For a precise definition of the *signature* of a channel, consider a channel of length L with some main coefficient $h_0 > 0$ the interference coefficients $h_1, \ldots, h_{L-1}$. For the ease of representation, the channel response is assumed causal. However, all the results of this section can be trivially reformulated to include the precursor interference coefficients. Formally, the channel signature $\mathbf{s}$ can be formulated as

$$\mathbf{s} = (s_0, \ldots, s_{L-1}) \in \{-1, 1\}^L, \text{ where } s_i = \text{sign}\,(h_i) \text{ for all } \; i = 0, \ldots, L-1.$$

Then, the worst-case interference occurs on symbol X_i if and only if

$$(X_i, X_{i-1}, X_{i-2}, \ldots, X_{i-L+1}) = \pm(s_0, -s_1, -s_2, \ldots, -s_{L-1}).$$

To further streamline the notation, let $\mathbf{p} = (s_0, -s_1, -s_2, \ldots, -s_{L-1})$ so that $\mathbf{p}$ and $-\mathbf{p}$ denote the two symbol patterns that cause the worst-case interference on the most-recently transmitted symbol. In other words, when $(X_i, \ldots, X_{i-L+1}) = \mathbf{p}$, symbol X_i incurs the worst-case ISI. The notation differs from that employed in the previous section since it is now convenient to include the most recently transmitted symbol, X_i, into the symbol patterns.

To illustrate the extent to which the channel response controls the joint error statistics in the worst-case-dominant regime, the following lemma shows that if two consecutive symbols can jointly be in error, the corresponding channel response must have one of two possible signatures.

Lemma 1. *If $f = 1$ and some two consecutive symbols are found to be in error, then the channel signature is either given by*

$$\mathbf{s} = \pm(1, -1, -1, -1, \ldots, -1)$$

or by

$$\mathbf{s} = \pm(1,\ 1, -1,\ 1, -1, \ldots, (-1)^{L-2}).$$

Proof. Let the errors occur on symbols indexed i and $i-1$. Due to the symmetry, it is sufficient to assume $X_i = 1$. Let $X_{i-1} = X_i$. Since $f = 1$, it follows that $(X_i, \ldots, X_{i-L+1}) = \mathbf{p}$ and $(X_{i-1}, \ldots, X_{i-L}) = \mathbf{p}$. Matching the terms in common, it follows that $p_0 = p_1, p_1 = p_2, \ldots, p_{L-2} = p_{L-1}$. Thus, $\mathbf{p} = (1, 1, \ldots, 1)$ or $\mathbf{p} = (-1, -1, \ldots, -1)$, which corresponds to the channel signature $\mathbf{s} = \pm(1, -1, -1, \ldots, -1)$. Similarly, letting $X_{i-1} = -X_i$ requries that $p_0 = -p_1, p_1 = -p_2, \ldots, p_{L-2} = -p_{L-1}$. Thus, $\mathbf{p} = \pm(1, -1, 1, -1 \ldots, (-1)^{L-1})$, which corresponds to the channel signature $\mathbf{s} = \pm(1, 1, -1, 1, -1, \ldots, (-1)^{L-2})$. □

Since admitting the occurrence of multiple errors in a block of n symbols imposes a certain structure on the communication channel when $n < L$, it is convenient to develop some indicator of when, for a given channel, multiple errors in a block can occur. For this purpose, consider an indicator function c given by

$$c\,(l, \mathbf{p}) = \frac{1}{L-l} \left| \sum_{j=l}^{L-1} p_{j-l} p_j \right|, \tag{2.21}$$

where $\mathbf{p} \in \{-1, 1\}^L$ is a worst-case symbol pattern and $l = 1, \ldots, L-1$ is the "delay". Due to its resemblance to the usual autocorrelation function, and for the lack of more a sensible terminology, the function c is referred to as the *pattern-correlation function* for a given channel. The following theorem provides the motivation for the above definition.

Theorem 4. *Consider an unconstrained stream of symbols transmitted over some channel of length L with the corresponding worst-case patterns $\pm\mathbf{p}$. Assume that $f = 1$ and consider symbols X_i and X_j for some $i, j \in \mathbb{Z}$, $i > j > i - L$. Then, errors can occur jointly on both symbols with some non-zero probability if and only if $c\,(i - j, \mathbf{p}) = 1$.*

Proof. Let $j = i - l,\ 1 \leq l \leq L-1$ and write the stream of transmitted symbols affecting Y_i and Y_j as

$$\begin{array}{cccccccc} X_i & X_{i-1} & \cdots & X_{i-l} & X_{i-(l+1)} & \cdots & X_{i-(L-1)} & \\ & & & X_j & X_{j-1} & \cdots & X_{j-(L-l-1)} & \cdots & X_{j-(L-1)} \end{array}$$

Since $f = 1$, errors occur on Y_i and Y_j if and only if $(X_i, \ldots, X_{i-L+1}) = \pm\mathbf{p}$ and $(X_j, \ldots, X_{j-L+1}) = \pm\mathbf{p}$. But, because $X_{i-l} = X_j,\ X_{i-(l+1)} = X_{j-1},\ \ldots,\ X_{i-L+1} = X_{j-(L-l-1)}$, then either $(p_0, \ldots, p_{L-l-1}) = (p_l, \ldots, p_{L-1})$ or $(p_0, \ldots, p_{L-l-1}) = (-p_l, \ldots, -p_{L-1})$. Either way, $c(i-j, \mathbf{p}) = 1$. Conversely, if $c(i-j, \mathbf{p}) < 1$, then either X_i or X_{j-1} is not in the worst case. □

One can now distinguish two cases: one where, for a given channel with worst-case patterns $\pm\mathbf{p}$, $c(l, \mathbf{p}) < 1$ for all $l = 1, \ldots, L-2$ and another where $c(l, \mathbf{p}) = 1$ for some $l = 1, \ldots, L-2$. In the former case, the channel is referred to as being *uncorrelated*, while, in the latter, the channel is considered to be *correlated*. Note that, trivially, $c(L-1, \mathbf{p}) = 1$ for any worst-case pattern $\mathbf{p}$.

Uncorrelated Channel

The principal result concerning uncorrelated channels is a direct corollary of the previous theorem.

Theorem 5. *If $f = 1$ and $c(l, \mathbf{p}) < 1$ for all $l = 1, \ldots, L-2$, then for any $n < L$, the event of observing two or more errors in a block of n symbols occurs with probability zero.*

Proof. For an uncoded system, since the set of possible transmitted symbol patterns is unconstrained, the result follows immediately from Theorem 4. For a coded system, the result follows by applying Theorem 1 prior to applying Theorem 4. □

The above result is of significant use for linear block codes since it implies that, in the limit $f \to 1$, the single parity check code reduces the error probability to zero. However, in practice, it is sometimes difficult to measure all, or any, of the channel

coefficients with certainty, and thus the full channel signature may not be known. The following theorem shows that for some fixed codeword length n and assuming all possible channel signatures to be equally likely, the probability of observing an uncorrelated channel tends to unity with increasing channel length.

Theorem 6. *Let the vector random variable* $\mathbf{P}$ *take on a value from the set* $\{-1,1\}^L$ *according to a uniform distribution. Then, for some fixed integer* n *such that* $1 < n < L$ *and any* $l = 1,\ldots,n$,

$$P(c(l,\mathbf{P}) = 1) \to 0 \text{ as } L \to \infty.$$

Proof. Observe that

$$\mathrm{P}(c(l,\mathbf{P}) = 1) = \frac{2^{l+1}}{2^L} \leq 2^{n+1-L}$$

Thus, $\mathrm{P}(c(l,\mathbf{P}) = 1) \to 0$ as $L \to \infty$. □

Correlated Channels

Now consider an unconstrained sequence of symbols transmitted over a correlated channel. Observe the sequence of symbols $X_i,\ldots,X_{j-L+1}$, for some $i,j \in \mathbb{Z},\ i > j > i-L$, and denote by p_{err} the probability of a detection error occurring on symbol X_i. Let $l = i-j$. Then, assuming that $c(l,\mathbf{p}) = 1$, the probability of observing an error jointly on X_i and X_j becomes $p_{err}^2 \times 2^{L-l-1}$. This result is obtained noticing that, when $(X_i,\ldots,X_{i-L+1}) = \pm\mathbf{p}$, there are only l free symbols preceding X_j.

To further consider the effect of the joint statistics on a coded system, define the *minimum correlation distance* $\lambda \in \{1,\ldots,L\}$ as the distance to the first correlated index, that is, $c(\lambda,\mathbf{p}) = 1$ and $c(l,\mathbf{p}) < 1$ when $l < \lambda$. The correlation distance becomes a practical quantity when implementing error-correction codes in the worst-case-dominant regime. For instance, assuming $f = 1$, a single parity check code will sucesfully correct all the errors as long as $n < \lambda$.

Instead of pursuing these and similar issues further, the analysis of the joint er-

ror behavior in correlated channels is postponed until Section 2.4, which develops guidelines regarding error-control coding in high-speed links. Note that the need to communicate adequately over correlated channels provides the motivation for the *pattern-eliminating codes*, also introduced in that section.

Extensions to Quasi-worst-case Scenarios

For the case of a dichotomous channel operating in a quasi-worst-case scenario, the previous results generalize trivially[11] as long as the coefficients in the principal part of the response are consecutive. The case where the latter are not consecutive is anologous to a channel with some zero-magnitude coefficients operating in the worst-case-dominant regime. Intuitively, allowing zero-magnitude coefficients translates to "don't care" positions in the worst-case patterns. By nulling out channel coefficients, it thus becomes easier to nest worst-case patterns and the errors effectively become more correlated. However, instead of pursuing this issue further, it is more relevant to consider it from the coding point of view, as done in Section 2.4.

2.3.2 Joint Error Behavior in the Large-set-dominant Regime

Experimental results, such as the simulation results of Chapter 3 or the measurements reported in [4], suggest that even in systems with inter-symbol interference, the symbol errors can often be considered *statistically independent.* This immediately holds in the large-noise scenario. For instance, consider a channel of length L with the interference coefficients on the order of $1/L$. Then, the variance of the ISI, given by $\frac{L-1}{L^2}$, vanishes in the limit of large L. Thus, for some given noise variance and sufficiently large L, the error events are effectively independent.

The validity of the independence assumption for the large-set-dominant scenario is not immediately obvious. Since the noise is considered independent across the sampling instants, potential dependencies are due to the interference component alone. For uncorrelated channels with interference coefficients of relatively even magnitude,

[11] The approach is the same as that of Section 2.2.1.

the following discussion shows that the error events in the large-set-dominant scenario can generally be considered independent. Note that, due to the practical focus of this thesis, the following arguments are formulated informally. However, the following lemma provides the rationale for the result.

Lemma 2. *Let* $\mathbf{x} = (x_i, \ldots, x_L)$ *and* $\mathbf{y} = (y_1, \ldots, y_L)$ *represent two L-dimensional vectors, where* $\mathbf{x}, \mathbf{y} \in \{-1, 1\}^L$ *and L is even. Given* $\mathbf{x}$*, let S denote the set of all vectors which match* $\mathbf{x}$ *in exactly* $L/2$ *positions, compared term-wise. Let* $\mathbf{y} \in S$*. It follows that the number of elements of S that match* $\mathbf{y}$ *in exactly m positions is given by* $C_L^m = \frac{L!}{m!(L-m)!}$ *when m is even, and is zero for m odd.*

Proof. First pick any $\mathbf{x}, \mathbf{x} \in \{-1, 1\}^L$ and note that the number of vectors in $\{-1, 1\}^L$ that match $\mathbf{x}$ in exactly l locations is given by C_L^l,

Now pick some $\mathbf{y}, \mathbf{z} \in$ S and let $\mathbf{y}', \mathbf{z}' \in \{0, 1\}^L$ be indicator vectors such that $y_i' = 1 \iff x_i = y_i$ and $z_i' = 1 \iff x_i = z_i$. It follows that, $y_i = z_i \iff y_i' = z_i'$. Notice that for $\mathbf{z}$ to mismatch $\mathbf{y}$ in an odd number of positions, either $\mathbf{y}$ or $\mathbf{z}$ would need to mismatch $\mathbf{x}$ in an odd number of locations, which contradicts the definition of S. Similarly, for $\mathbf{z}$ to match $\mathbf{y}$ in exactly m locations where m is even, $\mathbf{z}$ must match $\mathbf{x}$ in exactly m locations too. The number of possible vectors $\mathbf{z}$ that do so is C_L^m. □

Consider a channel of length L whose interference coefficients $h_1, \ldots, h_{L-1}$ are some fixed values taken from the set $\{-\delta, \delta\}$ with $\delta > 0$. Note that the probability of observing some interference value from the set $\{-(L-1)\delta, \ldots, 0, \ldots, (L-1)\delta\}$ is governed by a binomial distribution. First consider the limit where the error is due with probability 1 to the set of symbol patterns which mismatch the channel signature in exactly half of the positions. Suppose that the channel signature is uncorrelated, where the term is used in the usual sense[12]. Ignoring the effect of the edges, this implies that the shifted versions of the channel match in exactly half of the positions. By the above lemma, it then follows that, conditioned on the occurrence of a detection

[12]That is, not in the sense of pattern-correlation

error at time index i, the new probability distribution governing the interference value at time $i + L > j > i$ is still binomial for even multiples of δ. Although interference values at odd multiples of δ can no longer occur, the error probability remains roughly the same[13]. The errors at times i and j thus *occur independently.*

The above result can be extended to practical large-set-dominant conditions, up to a certain accuracy. Still considering the uncorrelated channels with interference coefficients of approximately even weight, consider the case where the a posteriori probability mass is concentrated on a signal value other than the signal mean. Then, the above result still applies as long as the following two conditions hold

1. There are sufficiently many symbol patterns associated with that signal value.

2. The corresponding patterns have a roughly equal distribution of -1s and 1s.

The second condition ensures that the conditional probability distribution for any adjacent signal is sufficiently close to a binomial, while the first ensures that the gaps between the allowed conditioned signal values are relatively small. Note that, for a given signal value, the first condition improves with the increasing L, while a decreasing δ reduces the effect of unevenness in the distribution of -1s and 1s in a given pattern.

For uncorrelated channels with interference coefficients of approximately even weight, and provided that the above two conditions are satisfied, the independence assumption extends to many practical situations. For instance, these include the case discussed in Section 1.3.3 of Chapter 1, where low noise and minimum decision distance can lead to the a posteriori probability mass concentrating on the signal values in the vicinity of the decision threshold. The same holds when the probability mass is concentrated at multiple such points, and the result thus extends to arbitrary large-set-dominant scenarios. It is also interesting to observe the cases where the independence assumption fails. As expected, it does not hold in the worst-case-dominant

[13]The change in the error probability due to "deleting" half of the possible interference positions is a factor of the magnitude of δ and the noise standard deviation σ. When δ is on the order of σ or less, the discrepancy is negligible for practical purposes. This is the only case of interest since the error probability is otherwise unrealistically high (since the channel is moderately long and interference coefficients roughly uniform).

regime, regardless of the channel coefficients, since condition 1) fails. Although the two conditions can be satisfied in the quasi-worst-case-dominant case, the fact that the interference coefficients are not all of approximately equal weight renders the argument invalid. Specifically, conditioned on the occurrence of an error on a given received signal, the symbols corresponding to the location of the principal channel coefficients are tied to their worst-case values. Then, considering the received signal at the next sampling instant, the likely symbol patterns are no longer binomially distributed — they are tied to some pattern formed by the symbols in the principal part. The above results are summarized in the following proposition.

Proposition 1. *Consider a system operating in a large-set-dominant regime. For an uncorrelated channel of some large length L and interference coefficients $h_1, \ldots, h_{L-1}$ of relatively even magnitude, the error events are approximately independent, regardless of the noise variance. The approximation improves with the increasing L and/or decreasing coefficient magnitude, as long as the regime is preserved.*

The above conditions apply, for instance, to high-speed link channel equalized for dispersion where inter-symbol interference is due to multiple, significantly attenuated signal reflections. Note that, for long channels in general, pattern-correlation roughly equals the usual correlation. By Theorem 6, a very large proportion of channel signatures of length L is in fact uncorrelated.

Although it is difficult to accurately quantify the length of the channel or the distrbution of the interference coefficients necessary for the independence assumption to hold with sufficient accuracy, the simulation results of Section 3.5 of Chapter 3 suggest that the independence assumption is relatively accurate for a range of conditions. Note, however, that the above argument assumes the transmitted symbols to be unconstrained and the channel coefficients to have fixed magnitude. The impact of the code constraints is also quantified in Section 3.5 in the context of system simulation for coded high-speed links.

2.4 Codes for ISI-and-AWGN-limited Systems

Based on the results of Sections 2.2 and 2.3, this section provides a set of guidelines regarding the adequacy of different codes for given interference and noise scenarios. It begins by discussing the benefit of classical error control codes, which are shown to be optimal in some regimes. However, the main body of the results serves to develop new codes, better suited for correlated channels in the worst-case-dominant (or quasi-worst-case-dominant) regime.

2.4.1 Classical Error-Control Codes

From the performance point of view, since the classical coding theory provides an exhaustive characterization of different error-control codes, the cases where the latter are a suitable choice require little additional analysis. Further work, however, is necessary to identify optimal error-control codes from the standpoint of the complexity/performance tradeoff, critical in high-speed links. Classical coding theory applies in the large-noise scenario, as interference has sufficiently little effect on both the symbol error probabilities and error dependencies. By the results of Section 2.3.2, the same generally holds for uncorrelated channels in the large-set-dominant scenario.

Considering error-control codes in a worst-case-dominant regime yields some suprising results. Referring to the results of Section 2.3.1 for an *uncorrelated* channel where, in the limit as $f \to 1$, at most one detection error occurs per codeword of length n, the optimal code is the *single parity check code* (SPC). Specifically, for an uncorrelated channel, the probability of observing multiple bit errors separated by $L-1$ symbols or less vanishes in the limit. Since the SPC corrects all single-bit errors in a codeword, the error rate after decoding therefore also vanishes, as long as $n \leq L-1$. In practice, the probability of multiple errors in a codeword over an uncorrelated channel drops off rapidly as f approaches unity. Moreover, on some channels, the probability of multiple errors in a codeword is additionaly reduced through possible *pattern-eliminating properties* of the SPC. In other words, since the SPC reduces the set of allowed previously-transmitted symbol patterns by roughly a factor of $2^{\lfloor L/n \rfloor}$,

it is likely to reduce the occurrence of harmful symbol patterns by the virtue of constraining the symbol stream. These observations also extend to dichotomous channels in the quasi-worst-case scenario by considering only the principal portion of the channel. It follows that the SPC may provide an efficient tool for combatting the ISI for uncorrelated channels at realistic values of f, in the worst-case or quasi-worst-case-dominant regime.

For *correlated channels* in the worst-case or quasi-worst-case-dominant regime, an occurrence of a detection error renders errors on the surrounding symbols more likely. Regarding the performance of the SPC, the error probability still vanishes as long as $n < \lambda$, where λ is the channel's minimum correlation distance defined in Section 2.3.1. However, for small λ, letting $n < \lambda$ can significantly reduce the code rate.

Since, for heavily correlated channels in these regimes, multiple-error occurrences within one codeword become more likely, standard error-correction codes are generally impractical due to the large necessary overhead. Furthermore, common techniques for decoupling the error occurrences, such as interleaving, heavily tax the system's power budget. Specifically, the potentially long digital registers incur too much power and area overhead at typical high-speed link data rates[14]. Instead, for correlated channels, there is a more efficient method of error control as it becomes more advantageous, from the point of view of coding overhead, to focus on eliminating the occurrence of the worst-case symbol patterns. This gives rise to a new type of block codes, descriptively termed the *pattern-eliminating codes*. The remainder of this section focuses on these codes exclusively.

2.4.2 Pattern-eliminating Codes

Results of Section 2.2.1 show that if code constraints prohibit the worst-case patterns from occuring on a given symbol, the error probability vanishes as $1 - f$ when $f \to 1$. The natural question is thus whether one should employ codes to accomplish this purpose specifically. This section develops properties of $(n, n - c)$ systematic block codes whose sole function is to prevent the worst-case patterns from occuring on

[14]In fact, for these reasons, the high-speed links operate principally in the mixed-signal domain.

any information symbol. For obvious reasons, these codes are referred to as *pattern-eliminating codes.*

Pattern-eliminating codes differ from linear block codes in several important ways. The most immediate difference is that the pattern-eliminating codes do not operate by correcting or detecting errors, but by *preventing* errors through the elimination of harmful symbol patterns. Also, such codes are not necessarily linear. However, a more practical distinction between the two types of codes is the fact that the pattern-eliminating codes *require no decoding.* Specifically, since the codes are systematic and the constraint symbols are typically of little use for any redundancy checking, "decoding" simply consists of removing the constraint symbols from the information stream upon detection of the codeword. This is facilitated by the fact that the constraint symbols occur at pre-determined locations. The synchronization issues are the same as those encountered with linear block codes.

The Principle

In the context of this thesis, pattern-eliminating codes are $(n, n-c)$ systematic binary block codes, where c is the number of constrained symbol locations per block of n. For the ease of representation, it is assumed that the c constraint symbols are consecutive. The assumption matters little in situations where the channel coefficients are all of roughly equal magnitude. However, as further discussed in Section 2.4.3, there may be a benefit to spacing the constrained locations for some practical channels. This generalization can be part of the future work on pattern-eliminating codes for high-speed links.

Assuming the constrained locations to be consecutive, the constraint symbols are then necessarily transmitted first since it is otherwise difficult to have control over the symbol patterns affecting the received signal. However, the c constraint symbols depend not only on the $n-c$ information symbols that follow them, but also on the $L-(c+1)$ symbols that precede them. In other words, the c constraint symbols are chosen based on the fully known history affecting each of the $n-c$ information symbols. This choice allows the most efficient use of the c constrained locations.

Specifically, considering the full transmit history reduces the set of goals that the c constraint symbols need to achieve and thus either allows for the minimum value of c to be used or, for a given c, provides means of eliminating additional error-causing patterns. The latter option is particularly attractive for cases where $f < 1$, yet f is large enough so that a limited number of symbol patterns causes the majority of the errors. The following example illustrates the limitations of constraining the c symbol locations based only on the current $n - c$ information symbols.

Example 1 *Consider a channel of length $L \geq 3$ with the worst-case patterns $\pm(p_0, \ldots, p_{L-1})$. Assume $n \geq 3$ and let $X_n, \ldots, X_1$ represent the codeword symbols with X_1 denoting the single constraint symbol, transmitted first. Suppose that, while setting the value of the constraint symbol given some input pattern. one ignores all symbols transmitted prior to X_1. Then, to ensure that the information symbol X_2 is not preceded by a worst-case pattern, it is necessary that $(X_2, X_1) \neq \pm(p_0, p_1)$. Similarly, to ensure that X_3 is not preceded by a worst-case pattern either, it is also necessary that $(X_3, X_2, X_1) \neq \pm(p_0, p_1, p_2)$. If $p_0 = p_1 = -p_2$, this is impossible to achieve. Thus, the code is not able to eliminate all occurrences of the worst-case pattern. In order to render the code effective, it is necessary to increase the number of constraint symbols. It follows that the penalty of encoding based on n symbols alone is the potentially large increase in overhead.*

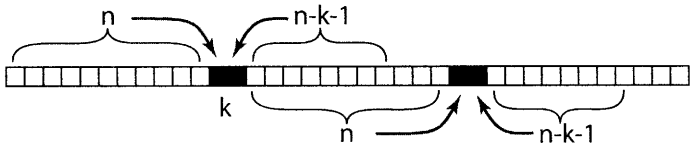

Figure 2-2: "Encoding" in Pattern-eliminating Codes

Concerning adequate codeword lengths, first note that choosing $n > L$ reduces the effectiveness of the code since at least one of the constraint symbols will have no effect on at least one of the information symbols. On the other hand, letting $n < L$ may either increase the effectiveness of the code in eliminating additional harmful patterns or simply decrease the stress on the constraint symbols. The latter may, in turn, lead to simpler encoder implementations, allow for a reduced c, or extend the set of channels for which an $(n, n - c)$ code is effective. The results of the following

two sections, which develop properties of $(n, n-1)$ and $(n, n-2)$ pattern-eliminating codes, are derived letting $n = L$ in order to provide the most restrictive conditions under which a pattern-eliminating code yields a benefit. The results concern the limit $f \rightarrow 1$, in which context optimality translates into the ability to prevent the worst-case patterns from preceding any of the information symbols. The remaining sections extend these results to a more practical context.

Pattern Elimination With a Single Constraint Symbol

The analysis of the pattern-eliminating codes starts by answering the question of when *a single constraint symbol*, i.e. $c = 1$, is sufficient to prevent all occurrence of the two worst-case patterns, $\pm\mathbf{p}$ employing the notation of Section 2.3, with respect to the $n - c$ information symbols.

All the results concerning the pattern-eliminating codes are obtained through the one-to-one correspondence between the channel signature and the two worst-case symbol patterns, $\pm\mathbf{p}$, which also provided the foundation for the results of Section 2.3.1. The proofs proceed mainly by *nesting* the worst-case patterns and observing the implications on the structure of the channel. To *nest* two worst-case patterns is to stagger them by l symbols, where $l = 1, \ldots, n - c - 1$ and equate the terms which overlap, as illustrated in the figure below. The range of l reflects the fact that the ISI which affects the c constraint symbols is irrelevant, since these are discarded upon detection.

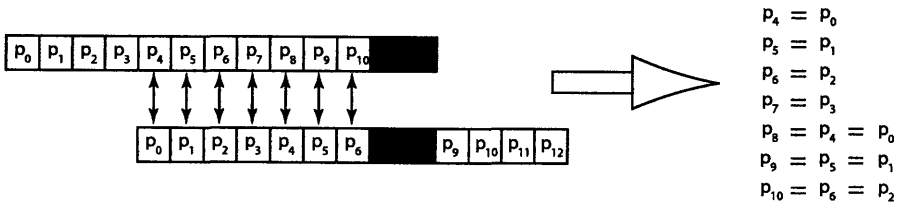

Figure 2-3: Nesting Worst-case Symbol Patterns

Prior to formulating a general criterion, the following example illustrates how an $(n, n-1)$ pattern-eliminating code can fail to be effective over a given channel.

Example 2 *Consider some channel response given by coefficients $h_0, h_1, \ldots, h_{L-1}$ and let the corresponding channel signature* $\mathbf{s}$ *be entirely positive, that is,* $\mathbf{s} = (1, 1, 1, \ldots, 1)$. *It follows that the worst-case patterns* $\pm\mathbf{p}$ *are given by* $\mathbf{p} = (1, -1, -1 \ldots, -1)$. *Let* $n = L$ *and consider the codeword symbols* $(X_n, \ldots, X_2, \tilde{X}_1)$, *where* $\tilde{X}_1$ *is the single constraint symbol whose value is set in some arbitrary fashion. Let*

$$(X_n, X_{n-1}, \ldots, X_2, \tilde{X}_1) = (1, -1, \ldots, -1, -1) = \mathbf{p}.$$

Then, it suffices to "toggle" the value of $\tilde{X}_1$ *to prevent the corresponding worst-case pattern from occurring. The transmitted codeword thus becomes,*

$$(X_n, X_{n-1}, \ldots, X_2, X_1) = (1, -1, -1, \ldots, -1, 1) \neq \pm\mathbf{p}.$$

However, given the above choice, consider the sequence of symbols $X_2, \ldots, X_{2-(n-1)}$, *where symbols* $X_0, \ldots, X_{2-(n-1)}$ *correspond to the first* $n-2$ *information symbols of the previously transmitted codeword. Then, if* $X_0, X_{-1}, \ldots, X_{2-(n-1)} = (1, 1, \ldots, 1)$, *it follows that*

$$(X_2, X_1, X_0, \ldots, X_{2-(n-1)}) = (-1, 1, 1 \ldots, 1, 1) = -\mathbf{p}.$$

Thus, when $X_1 = -1$, *the symbol* X_n *is affected by the worst-case interference, while when* $X_1 = 1$, *the worst-case interference affects symbol* X_2. *It follows that an* $(n, n-1)$ *pattern-eliminating code is ineffective on this channel.*

The following result provides the general criterion for determining whether, on a given channel, an $(n, n-1)$ pattern-eliminating code is effective.

Theorem 7. *Assume* $n = L$ *and let* $\mathbf{p} = (p_1, \ldots, p_n)$ *be a worst-case symbol pattern. Then, the following two statements are equivalent:*

(A): *An* $(n, n-1)$ *pattern-eliminating code is ineffective.*

(B): *The worst case patterns* $\pm\mathbf{p}$ *are such that there exist some* m *and* l *where* $1 < m < l \leq n-1$ *so that* $p_l = -p_m p_{l-m}$ *and* $p_{j'} = p_m p_{j'-m}$ *for all* $j' \neq l$, $m \leq j' \leq n-1$.

Proof Let $\mathbf{x}, \mathbf{y} \in \{-\mathbf{p}, \mathbf{p}\}$ be some two worst-case symbol patterns. Create a pattern $\mathbf{y}'$ by "toggling" exactly one symbol in $\mathbf{y}$. Then, if $\mathbf{x}$ and $\mathbf{y}'$ can be nested so that they are separated by at most $n-1$ symbols, the $(n, n-1)$ pattern-eliminating code is ineffective (a worst-case pattern occurs regardless of the value of the constraint symbol). Conversely, it is clear that this is the only way for the pattern-eliminating code to fail. The condition **(B)** directly follows by translating the previous statements into appropriate notation. □

The above result is in essence a search algorithm[15] of complexity $2n$. For instance, the channel of Example 1 fails with $l = n-1$ and $m = n-2$. Note, however, that the results of this section are derived under the most restrictive condition where $n = L$. As a general rule, since most of the proof techniques rely on nesting worst-case patterns, relaxing this condition further limits the extent to which different patterns can be nested and can thus improve the performance of a pattern-eliminating code. To illustrate this fact, reconsider the code of the previous example while adding one pre-cursor coefficient h_{-1} to the channel, while maintaining the codeword length constant.

Example 3 *Now consider the channel of length $L+1$ given by $h_{-1}, h_0, \ldots, h_{L-1}$ where $\mathbf{s} = (1, 1, 1, \ldots, 1)$ as previously. The worst-case patterns $\pm\mathbf{p}$ become:*

$$\mathbf{p} = (-1, 1, -1, -1 \ldots, -1).$$

Consider a pattern-eliminating code of codeword length $n = L$. Following the previous notation, assume that

$$(X_n, \ldots, X_2, \tilde{X}_1) = \mathbf{p}.$$

In this case, note that it still suffices to "toggle" the value of $\tilde{X}_1$ to prevent the corresponding worst-case pattern from occurring. However, due to this particular channel signature and the fact that, while nesting symbol patterns, the pre-cursor symbol needs to be taken into

[15] Given some channel response, it suffices to compute the worst-case pattern $\mathbf{p}$. Then, by trying the $n-1$ possible nestings of $\mathbf{p}$ with $\mathbf{p}$ or $-\mathbf{p}$, one either finds some values for m and l, in which case the code is ineffective, or one does not, in which case the code is effective.

account as well, it is impossible to nest a second worst-case pattern that would lead to a contradiction. The code is therefore effective.

The above example can be generalized to show that for an all-positive channel of length L, an $(n, n-1)$ pattern-eliminating code is effective as long as $n < L$. This result is of practical use, as it indicates how an $(L-1, L-2)$ pattern-eliminating code can be successfully applied to dispersive channels. As another illustration, for the two channels of Lemma 1, an $(n, n-1)$ pattern-eliminating code is effective for any $n \leq L$. The result follows by noticing that the worst-case patterns always nest fully, which implies that it is impossible to derive a contradiction similar to those derived in the first example. These observations are formalized as the following corollaries.

Corollary 2. *For a channel length L with signature $\mathbf{s} = \pm(1, 1, 1, \ldots, 1)$ an $(n, n-1)$ pattern-eliminating code is effective as long as $n \leq L - 1$.*

Corollary 3. *For a channel of length L with signatures $\mathbf{s} = \pm(1, -1, -1, -1, \ldots, -1)$ or $\mathbf{s} = \pm(1, 1, -1, 1, -1, \ldots, (-1)^{L-2})$, an $(n, n-1)$ pattern-eliminating code is effective as long as $n \leq L$.*

Note that the effectiveness of an $(n, n-1)$ pattern-eliminating code is generally not tied to the channel correlation. If a channel is uncorrelated, then an $(n, n-1)$ pattern-eliminating code is ineffective for $n = L$, since the condition B of the previous theorem holds with $m = n - 2$ and $l = n - 1$. However, this may no longer be true for $n < L$. Similarly, it cannot be inferred that, in general, an $(n, n-1)$ pattern-eliminating code is effective on any correlated channel, although this happens to hold for the two correlated channels given in the Corollary 3. As a counterexample, consider a channel with the worst-case pattern $\mathbf{p} = (\ 1, -1,\ 1, -1, -1,\ 1, -1,\ 1, -1, -1)$. It is a matter of straightforward computation to establish that the channel is indeed correlated, but that condition B of Theorem 7 is satisfied, thus rendering the code unable to prevent all worst-case patterns.

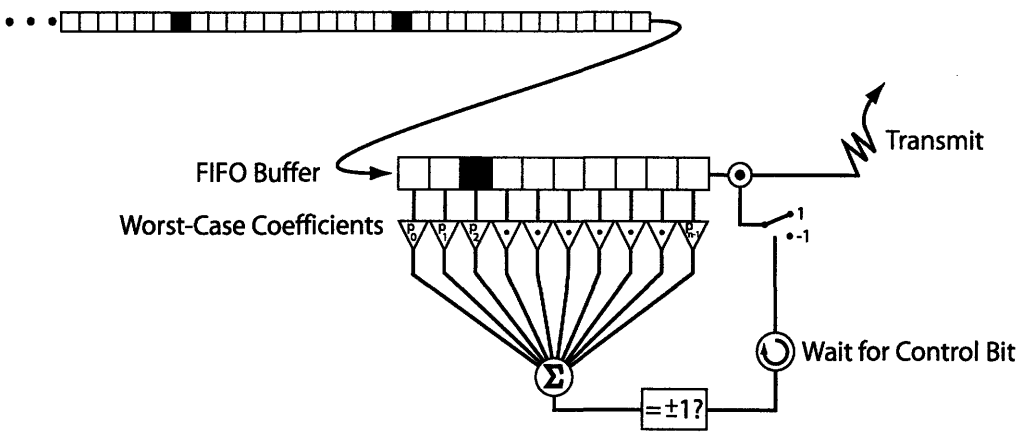

Figure 2-4: Possible Encoder Implementation for a $(n, n-1)$ Pattern-eliminating Code

Lastly, for channels where one constraint symbol is sufficient, the following result provides a simple algorithm for setting its value. It provides the basis for possible efficient implementations, one of which is shown in Figure 2-4.

Theorem 8. *Consider a channel over which an $(n, n-1)$ pattern-eliminating code is effective. Let $x_n, \ldots, x_2$ denote some realization of the $n-1$ codeword information symbols and $x_0, \ldots, x_{2-(n-1)}$ denote a realization of the $n-1$ previously transmitted symbols. Then, the optimal value of the constraint symbol x_1 can be determined as follows.*

Pick some $x \in \{-1, 1\}$. If setting $x_1 = x$ yields $(x_i, \ldots, x_{i-n+1}) \neq \pm\mathbf{p}$ for $i = 1, \ldots, n-1$, then x is the optimal choice. Otherwise, set $x_1 = -x$.

Proof. For any $x \in \{-1, 1\}$, if $(x_i, \ldots, x_{i-n+1}) \neq \pm\mathbf{p}$ for $i = 1, \ldots, n-1$, then x is the best possible value for x_i. Otherwise, since x can take only two values and since an $(n, n-1)$ code is effective for this channel, then the complement of x is necessarily the desired choice. □

Pattern Elimination On Any Channel

The following theorem, whose proof is included in the appendix, shows that *two* constraint symbols are sufficient for a pattern-eliminating code to be *effective over any channel.* However, due to the rate penalty of codes for bandwidth-limited systems,

further discussed in the following section, the issue of encoding for $(n, n-2)$ pattern-eliminating codes is not addressed further. Instead, the following sections focus on improving the performance of $(n, n-1)$ pattern-eliminating codes on channels where such codes are effective.

Theorem 9. *For any channel of finite length L and any $n \leq L$, the $(n, n-2)$ pattern-eliminating code is effective.*

Performance of Pattern-eliminating Codes

The previous section identified the channel signatures for which a pattern-eliminating code prohibits any occurrence of the worst-case symbol patterns. Given a system operating in the worst-case-dominant regime with some high f, the error probability of a pattern-eliminating code, p'_{err}, is thus upper-bounded by Theorem 3, that is,

$$p'_{err} \leq 2^c(1-f)p_{err},$$

where p_{err} is the error probability of the uncoded system operating at the same signalling rate. Note that a higher value of f can be obtained by considering the worst-case patterns formed only by the "significant" interference coefficients[16], whose magnitude is large compared to the standard deviation of the noise. This is equivalent to considering a dichotomous channel operating in the quasi-worst-case-dominant scenario. In fact, Sections 2.4.3 and 2.5 show that dispersive channels at typical high-speed link noise levels *operate in the quasi-worst-case-dominant regime with very high* f, on the order of $1 - 10^{-10}$, thus yielding error-rate reductions of over ten orders of magnitude compared to the uncoded case. It now remains to determine when, given a bandwidth-limited system, the benefit of the pattern-eliminating code exceeds its rate penalty.

Consider an $(n, n-c)$ pattern-eliminating code applied to the six communication

[16]For instance, for a dispersive channel whose ISI spans 200 symbols, there is typically a benefit in focusing only on eliminating the worst-case patterns defined only by the first few interference coefficients, i.e. the dispersion coefficients. The coefficients in the tail of the response are typically very small and the effect of an individual coefficient is negligible.

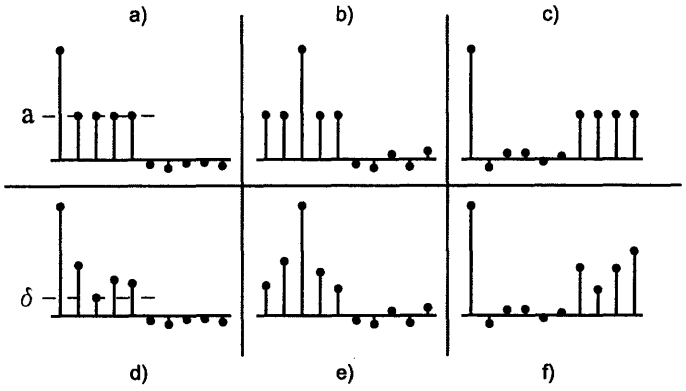

Figure 2-5: Channel Examples

channels illustrated in Fig. 2-5a)-f). The channels have the same signature, but differ by the position of the significant interference coefficients and by their respective magnitudes. Suppose that on channels with an all-positive signature, the code guarantees some d symbols of separation between allowed symbol patterns and the worst-case patterns when $n = L$. Denote the corresponding improvement in the minimum decision distance, guaranteed for any information symbol, by Δ_c, For channels a)-c) whose interference coefficients of equal magnitude a, the improvement is given by

$$\Delta_c = 2da \tag{2.22}$$

On the other hand, for channels d)-f) where the interference coefficient are of varying magnitude,

$$\Delta_c = \min(|h_{j_1}| + ... + |h_{j_d}|) \quad \text{where} \quad 1 \leq j_1 < ... < j_d \leq c. \tag{2.23}$$

The above expression shows that the optimal performance is derived when the interference coefficients of equal magnitude, keeping the total power constant. Otherwise, for some channels, there may be a benefit to "spacing" out the constrained locations within a block of n symbols so that the minimum is no longer taken over consecutive coefficients. Note, however, that $(n, n-1)$ pattern-eliminating code yields $\Delta_c = 2\delta$, where δ denotes the magnitude of the least coefficient. When δ is relatively small compared to other coefficients, the error on the corresponding symbol dominates the

error expression and, at high f, the $(n, n-1)$ pattern-eliminating code may not yield a sufficient improvement.

For systems where the rate penalty is not critical, the above difficulty can be circumvented by reducing the codeword length n until all interference coefficients are of some suitable magnitude. Note that it suffices that the significant interference coefficients are consecutive, and need not occur adjacently to the principal signal component, as shown in Fig. 2-5 c) and f). However, for typical bandwidth-limited systems, both the code rate and the performance improvement Δ_c need to be taken into account to evaluate the full effect of the code. For some uncoded system operating at rate R, let $R' = R\frac{n}{n-c}$ denote the equivalent coded data rate, so that the effective information rates of the coded and the uncoded system are equal. Let Δ_r denote the improvement in the minimum decision distance when decreasing the signalling rate from R to R'. Thus, for the pattern-eliminating code to yield a benefit compared to an uncoded system operating at a lower rate, it is necessary that

$$\Delta_c \geq \Delta_r.$$

From the point of view of a system designer, to determine whether to implement a pattern-eliminating code given a channel measurements at different rates, it suffices to identify potential values of n and c, based on the channel measurement at rate R. Then, computing the quantities Δ_c and Δ_r from the channel measurements at the corresponding rates R', the best $(n, n-c)$ pattern-eliminating code is the one with the greatest positive difference $\Delta_c - \Delta_r$. Naturally, if the difference $\Delta_c - \Delta_r$ is negative, the pattern-eliminating code does not provide a decision distance benefit over an uncoded system operating at the same information rate.

At this point, it is also relevant to compare the performance of the $(n, n-1)$ pattern-eliminating code to that of the SPC. For a channel of length L over which both codes are effective, the $(n, n-1)$ pattern-eliminating code incurs less overhead than the SPC. Specifically, both codes require a single constraint/redundancy symbol per codeword, but the maximum codeword length of the SPC is $L-1$, while that of

the pattern-eliminating code is L. The SPC is preferable for a subset of uncorrelated channels over which an $(n, n-1)$ pattern-eliminating code of comparable overhead is not effective. Conversely, an $(n, n-1)$ pattern-eliminating code, when effective, is preferable to the SPC over correlated channels, especially if the minimum correlation length renders the overhead of the latter impractical. In fact, the two fully-correlated channels of Corollary 3 provide an illustration for why classical error-control codes often require too large of an overhead on channels with ISI. Finally, further work is required to establish which code is preferable at modest values of f, over channels where both codes are effective and incur the same overhead.

Timing Benefits of Pattern-eliminating Codes

The concept of pattern-eliminating codes can be further extended to include additional timing-related benefits, such as run-length-limiting (RLL). This has the advantage of effectively reducing the rate penalty of the code. In particular, for systems where timing-related properties are needed and are thus allocated a certain overhead, the rate penalty of a pattern-eliminating code vanishes, provided the code can provide the necessary timing benefit. For instance, consider a dispersive channel with no pre-cursor coefficients (although the following result extends to the pre-cursor case as well). The worst case symbol patterns for such channel are of the form $\pm\mathbf{p}$ where $\mathbf{p} = (-1, 1, 1, \ldots, 1)$ and has length L. Now consider an L-symbol run of ones given by $\mathbf{r} = (1, 1, 1, \ldots, 1)$. Note that pattern $\mathbf{r}$ is sufficiently different from $\pm\mathbf{p}$ in that the encoder will not create $\mathbf{r}$ when trying to prohibit $\mathbf{p}$, nor would it form the pattern $\mathbf{p}$ by trying to prohibit $\mathbf{r}$. Note, in addition, that an $(n, n-1)$ code is effective in removing all patterns of type $\mathbf{r}$ of length n or longer[17]. The encoding algorithm of Theorem 8 can therefore be augmented by a second simple rule to prohibit all occurrences of patterns $\pm\mathbf{r}$ in addition to $\pm\mathbf{p}$. The result is a $d = 1$ pattern-eliminating code with a run-length-limiting property, allowing for a maximum run of $n-1$ ones (or negative ones). This observation is generalized as follows.

[17]This can be shown using the similar argument to that in the proof of Theorem 7 and Examples 2 and 3.

Theorem 10. *The following statements are equivalent:*

A) *The $(n, n-1)$ pattern-eliminating code can be augmented by a rule to additionally prohibit all runs of 1s (or -1s) of length n.*

B) *The channel's worst-case patterns $\pm\mathbf{p}$ are such that there exists no integer j, where $2 \leq j \leq n$, so that $p_j = -1$ and $p_i = 1$ for all $i \neq j$.*

Moreover, the additional rule is given by Theorem 8, where the patterns to be prohibited are given by $\pm\mathbf{r}$ and $\mathbf{r}$ is a sequence of 1s of length n.

Proof. First note that the $(n, n-1)$ pattern-eliminating code can systematically prohibit all runs of 1s of length n or longer if and only if it prohibits the occurrence of patterns $\pm\mathbf{r}$, given above. Setting the value of the constraint symbol to prohibit the occurrences of the pattern $\mathbf{r}$ creates patterns $\pm\mathbf{p}$ where $p_i = 1$ except at some index j, where $2 \leq j \leq n$. If $\mathbf{p}$ is also the worst-case pattern for the corresponding channel, the pattern-eliminating rule and the run-length-limiting rule are not compatible. Conversely, if such $\mathbf{p}$ is the channel's worst-case pattern, then in seeking to prohibit it, the encoder can create patterns $\pm\mathbf{r}$. □

From the above result, it follows that the channels on which the pattern-eliminating rule and the run-length-limiting rule are not compatible are limited to those having the worst-case patterns of the form $\pm(1,-1,1,1,1,\ldots), \pm(1,1,-1,1,1,\ldots,1), \ldots,$ $\pm(1,1,\ldots,1,1,-1,1)$. Thus, the pattern-eliminating code, augmented by the simple rule analogous to that of Theorem 8, provides a run-length-limiting benefit over a majority of the channels. In particular, employing standard notation [3], the $(n, n-1)$ pattern-eliminating code additionally provides the full capability of a $(0, n-1)$-RLL code[18]. Furthermore, note that although a pattern-eliminating code can be augmented by an RLL property, a standard RLL code may not have an automatic pattern-eliminating benefit. In fact, the only instance where prohibiting a run of n ones provides a systematic pattern-eliminating benefit is over channels where the corresponding worst-case pattern $\mathbf{p}$ contains a run of n ones (or negative ones). Thus,

[18]The first parameter indicates the *minimum* guaranteed runlength, while the second indicates the *maximum* tolerated runlength. The RLL codes in magnetic recording provide some minimum runlength as means of reducing the ISI. This is however not necessary in digital transmissions, which include the high-speed links, and the value of the first parameter is usually set to zero [18].

the advantage of replacing a standard RLL code by an equivalent $(n, n-1)$ pattern-eliminating code is the potentially-large additional performance improvement at no additional overhead. In addition, its low complexity renders an $(n, n-1)$ pattern-eliminating an all the more desirable alternative. A numerical comparison between the 8b/10b RLL/DC-balancing code [18], commonly implemented in high-speed links, and an $(n, n-1)$ pattern-eliminating code with the same overhead is included in Section 2.5.

Further Work: Extending Pattern Elimination

Four possible extensions to the previously defined concept of pattern-eliminating codes, bearing potentially-significant practical benefits, follow immediately from the previous developments. They are described presently. A fifth major avenue for further work, namely the channel conditioning for pattern-eliminating codes, is discussed in Section 2.4.3.

First, allowing zero-magnitude coefficients in the channel response for systems operating in the worst-case-dominant regime, or allowing the coefficients in the principal part of the channel to be non-consecutive for the quasi-worst-case regime, is of practical importance. To illustrate the underlying problem, first note that, observing some symbol at time i, if a zero-magnitude coefficient aligns with the location of the constraint symbol, the constraint has no effect on the ISI incurred by the symbol at time i. In addition, it becomes easier to nest different patterns, and thus the number of situations where the pattern-eliminating code can be "baffled" in principle increases. As a partial solution, for any given channel, one can define the notion of the distance to the first null and ensure that it is not exceeded by the codeword length. However, the reduction in the codeword length increases the rate penalty of the code. A more interesting approach consists of spreading out the location of the constraint symbols within a codeword based on the location of the nulls in the channel response. For instance, while two consecutive nulls in the principal part of the channel response render any $(n, n-2)$ pattern-eliminating code ineffective, allowing the constraint symbols to be non-consecutive may reverse this effect. Further work is

required to evaluate the effect of constrained locations on the code performance, as a function of both the location and number of channel nulls present.

Second, for practical cases where f may not be sufficiently large for a pattern-eliminating code to yield a benefit in the worst-case-dominant or the quasi-worst-case-dominant regime, there may be a benefit to extending the set of prohibited patterns. To motivate future work in that direction, the following provides a preliminary result concerning the ability of a pattern-eliminating code to remove *all occurrences* of all patterns distant from the worst-case pattern by at least d symbols or less. The result is derived under the most restrictive condition, that is, letting $n = L$. However, as previously discussed, letting $n < L$ may improve the effectiveness of the code.

Theorem 11. *Consider channel of length L and an $(n, n-c)$ pattern-eliminating code whose c constraint symbols are consecutive and are transmitted first. Then, if $n = L$,*

$$d \leq \lfloor 3c/4 \rfloor \text{ for } c \text{ even or } \lceil 3c/4 \rceil \text{ for } c \text{ odd.}$$

Proof. Let $X_n, \ldots, X_{c+1}$ denote the codeword information symbols and $X_c, \ldots, X_1$ denote the constraint symbols. Focus on symbols X_n and X_{c+1} and nest the corresponding n-symbol history patterns as follows:

$$\begin{array}{cccccccccc} X_n & X_{n-1} & \ldots & X_{c+1} & X_c & \ldots & X_1 & & & \\ & & & X_{c+1} & X_c & \ldots & X_1 & X_0 \ldots & X_{c-n+2} \end{array}$$

Let $(X_n, \ldots, X_1) = \mathbf{p}$ and assume without loss of generality that $p_{n-c-1} = p_0$. Let $X_{c+1} = p_0$ and let $(X_{-1}, \ldots, X_{c-n+2}) = (p_{c+1}, \ldots, p_{n-1})$. In other words, denoting by $*$ the values at the constrained locations, the symbol streams are the following:

$$\begin{array}{cccccccccc} p_0 & p_1 & \cdots & p_{n-c-1} & * & \cdots & * & & & \\ & & & p_0 & * & \cdots & * & p_{c+1} \cdots & p_{n-1} \end{array}$$

Now set the value of constraint symbols optimally, ignoring any resulting interference on symbols other than X_n and X_{c+1}. Suppose $(p_1, p_2, \ldots, p_c)$ and $(p_{n-c}, \ldots, p_{n-1})$

are equal in j locations, set those symbols to be the opposite of their worst-case values and distribute the remaining symbols evenly between benefiting X_n and X_{c+1}. It follows that, for this information pattern, the two symbols can both be jointly distanced away from the worst case by at most $j + \lfloor (c-j)/2 \rfloor$ symbols.

Next, toggle the value of X_{c+1} and those of $X_0, \ldots, X_{c-n+2}$. Since $p_0 = p_{n-c-1}$, $(X_n, \ldots, X_1)$ is automatically distant from the worst case by one location. Since $(-p_1, -p_2, \ldots, -p_b)$ and $(p_{n-c}, \ldots, p_{n-1})$ are equal in $c-j$ locations, it follows that the two symbols can both be jointly distanced away from the worst case by at most $c - j + \lceil j/2 \rceil$ locations[19].

It follows that the *guaranteed* improvement over the $n-c$ information symbols is at most the minimum of the above quantities. The maximum of the resulting expression is reached when either $j = \lfloor c/2 \rfloor$ or $j = \lceil c/2 \rceil$. The result then follows by substitution.

□

As a consequence of the above theorem, letting $n = L$, it follows that for channels where an $(n, n-1)$ pattern-eliminating code is effective, an $(n, n-2)$ code provides the same pattern-eliminating benefit at an increased overhead and encoding complexity. In the case of a $(n, n-2)$ pattern-eliminating code, a previous result (Theorem 9) shows that the code does achieve the minimum guaranteed improvement of $d = 1$. However, the above theorem provides an easily-derived bound which may generally not be tight. In other words, although it upper-bounds the resulting improvement, it provides no information about the *guaranteed* improvement. It nevertheless is a useful lower bound on the amount of overhead necessary to achieve a certain performance. Specifically, to guarantee a certain distance of d symbols away from the worst-case pattern, the corresponding number of constraint symbols is on the order of $4d/3$. This suggests that for relatively short codewords, a potentially large overhead may be required when wishing to eliminate patterns from a larger set. It nevertheless remains that Theorem 11 was derived under the strictest condition, letting $n = L$. Although for a given c, reducing the codeword length to $n < L$ also increases the coding overhead, it is possible that the latter may be used more effectively.

[19]Since $\min\{c - j + \lfloor j/2 \rfloor + 1\ ,\ c - j + \lceil j/2 \rceil\} = c - j + \lceil j/2 \rceil$.

Thirdly, it remains to explore different timing properties and provide a more general characterization of the conditions under which these are compatible with the pattern-eliminating properties in an $(n, n-c)$ code. Moreover, there may be a benefit to further investigating the ties between the pattern-eliminating codes and the constraint codes for magnetic recording systems. Although, as discussed in Section 2.1.3, the two types of codes have largely different structures, some of the resulting insights pertaining to constraint coding over binary channels may be of use.

Finally, for $(n, n-c)$ pattern-eliminating codes with $c > 1$, an efficient encoding algorithm remains to be demonstrated. As the encoding is function of the constraint symbol location, the pattern-eliminating properties of the code and any timing benefit of the code, the previous avenues need to be explored prior to addressing this issue. It nevertheless remains that the simplicity of the encoding algorithm is one of the limiting factors for many applications. There is therefore a benefit to investigating simple, possibly sub-optimal, encoding algorithms. In particular, since some run-length-limiting codes allow for simple implementations, the previous body of work on coding for magnetic channels may provide some insight into this topic.

2.4.3 Coding for High-speed Links

This section revisits the previously developed results on coding for ISI-and-AWGN-limited systems in the context of high-speed links. As previously discussed, the question of coding in regimes where the effect of the inter-symbol interference is negligible has already been extensively addressed, although further work is required to characterize the performances of different coding schemes with respect to their hardware complexity. This applies to the large-noise regime and, as discussed in Section 2.3.2, to a large subset of the large-set-dominant regime. Specifically, for long uncorrelated channels with interference coefficients of relatively even magnitude, the error events on distinct symbols can be considered to be independent with some accuracy. Furthermore, the approximation improves with the increasing channel length and/or decreasing coefficient magnitude, as long as the regime is preserved. In high-speed links, the inter-symbol interference typically spans hundreds of symbols due to signal

reflections. In addition, for well-equalized channels with the dominant dispersion taps removed, individual coefficients are relatively small, although their cummulative effect can be large. Since the resulting pulse response is also uncorrelated, the previous result suggests that the errors can be considered[20] to occur independently. Then, the performance of classical error-control codes can be analyzed in the usual context of coding theory, with added attention to the issues of overhead and encoder/decoder complexity.

The pure worst-case-dominant scenario is impractical for high-speed links for many reasons. First, the reflected copies of the signal suffer more and more attenuation as they bounce accross the backplane and the line cards. Thus, the corresponding channel coefficients get progressively smaller in the tail of the channel response. Thus the noise levels required to achieve a sufficiently high value of f to observe a significant coding benefit with a pattern-eliminating code are unrealistically low. The same holds for the single-parity check code. Another important difficulty of coding for the worst-case-dominant scenario in high-speed links is notion of a "meaningful" impulse response. Since the system operates at low error rates, channel coefficients of otherwise-negligible magnitude can no longer be discarded, as their cumulative effect has a large impact on the tails of the error distribution. However, it is often difficult to determine whether some portions of the channel response are due to delayed symbols, like the commonly occuring reflections off of impedance discontinuities, or simply an artifact of the measurement. Both an artificial increase in the channel length and an artificial reduction of δ are harmful in that they falsely reduce the occurrences of the worst-case-dominant scenarios.

However, the greatest benefit of coding for high-speed links is in fact achieved for *dispersive channels in the quasi-worst-case-dominant scenario.* For such channels, both of the previously described difficulties disappear. Specifically, as further illustrated in the following section, the magnitude of the dispersion coefficients is

[20]More work is needed to quantify the accuracy of this assumption at very low error rates for the non-idealized scenario and constrained symbol alphabets. In the meanwhile, Section 3.5 of Chapter 3 illustrates this concept on a realistic high-speed link channel and error rates observable by Monte-Carlo simulation.

typically so large compared to the noise that the system operates in a quasi-worst-case regime with *very large* f. In addition, there is little ambiguity in measuring the polarity of the large channel coefficients that form the principal part of the channel. Furthermore, neither the exact values of the coefficients in the secondary part of the channel nor the corresponding channel length matter, as long as their cummulative effect is sufficiently small compared to the dispersion coefficients. It follows that, in dispersive high-speed link channels, which become more prominent as the signalling rates increase, the pattern-eliminating codes can provide a very large benefit. It now remains to provide a more precise description of the conditions under which a pattern-eliminating code on a dispersive channel yields a significant benefit.

Operating Conditions for Pattern-eliminating Codes

From the previous discussion it follows that relative to the noise variance, the larger the smallest coefficient in the principal part of the channel is, the larger the corresponding value of f. In addition, as large dispersion coefficients are a byproduct of high signalling rates over bandwidth-limited channels, it may appear that, armed with a low-overhead pattern-eliminating code alone, high-speed links can tolerate arbitrarily high data rates. However, as previously discussed in Chapter 1, the quasi-worst-case-dominant regime will not apply if the worst-case interference caused by the principal part of the channel, ignoring the secondary part, is sufficient to bring the signal over the decision threshold. This case is illustrated in Figure 2-6.

Exploring different ways of coping with the previously-described difficulty through some form of channel conditioning leads to the notion of *optimal channel equalization for pattern-eliminating codes.* More precisely, the standard use of channel equalization methods in high-speed links imposes a large demand on the corresponding equalizer, since the latter is typically the sole method of achieving the low error rates. On the other hand, when the equalizer is implemented jointly with a pattern-eliminating code, the goal of the former now becomes sufficiently reducing the interference for the minimum decision distance to become positive, although not necessarily very large. Thus the equalization burden is reduced, and with it the power consumption of the

high-speed link, and a simple pattern-eliminating code can now provide the required bit-error-rate improvements.

However, the problem of determining an efficient channel-conditioning method enabling the pattern-eliminating codes to yield a benefit at high data rates remains open. Revisiting the channel of Figure 2-6, the sufficient solution in this case consists of nulling the last dispersion coefficient, thus ascribing the latter to the secondary part of the channel without rendering the coefficients in the principal part non-consecutive. This is illustrated in Figure 2-7. Effectively nulling a channel coefficient can be achieved through a decision feedback equalizer (DFE) [3]. However, note that this increases the coding overhead, as the length of the primary channel is reduced by one symbol. A more significant problem inherent to the use of the DFE for channel conditioning in high-speed links is the fact that introducing more than three or four nulls in the channel pulse response is typically not energy-efficient, due to both the hardware overhead and the severe timing requirements on the corresponding structure.

Another common alternative to the DFE is a linear equalizer, commonly based on the zero-forcing criterion (ZFE) [3]. Due to its feed-forward structure and reduced timing demands, the linear equalizer appears to be a more efficient method of reducing the overall interference, especially when the corresponding reductions need not be large. However, by seeking to reduce each interference coefficient, the equalizer may introduce undesirable nulls in the principal portion of the channel's pulse response. Another serious problem associated with deploying linear equalization in a high-speed link relates to the latter's peak power constraint. More precisely, the requirement that the transmitted symbol power does not exceed some maximum given value has a detrimental effect on the signal to noise ratio of the technique and thus reduces the effectiveness of the linear equalizer. This is due to the fact that by normalizing the total power in the equalizer coefficients, one effectively reduces the power in the received signal, discounting the interference, compared to the equivalent unequalized system.

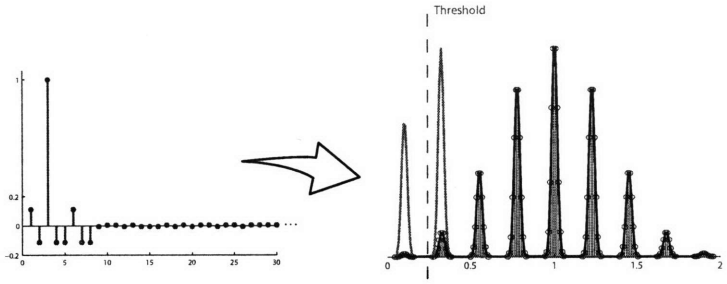

Figure 2-6: A high-speed link *not operating* in the quasi-worst-case-dominant regime – Note that the worst-case interference created by the principal part of the channel crosses the decision threshold.

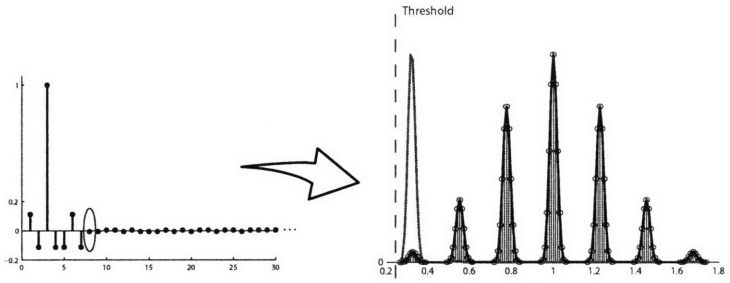

Figure 2-7: A high-speed link operating in the quasi-worst-case-dominant regime – By nulling the last interference coefficient, the worst-case interference created by the principal part of the channel no longer crosses the decision threshold.

Uncovering an efficient channel-conditioning method, as part of the further work on coding for high-speed links, would likely yield an increase in the achievable high-speed link data rates or the corresponding energy efficiencies, or both. In the meanwhile, Section 2.5 illustrates the potential of jointly optimizing a simple linear equalizer with an $(n, n-1)$ pattern-eliminating code over a realistic high-speed link channel.

2.5 Practical Examples

This section considers a few practical examples to illustrate the previously-developed ideas. However, note that the large-noise regime both forms the backbone of the communication theory, while the worst-case-dominant scenario seldom occurs in a high-speed link with sufficiently high f. Furthermore, the large-set-dominant regime is illustrated in Chapter 3 in the context of numerical simulation for high-speed links. The present section thus focuses on the quasi-worst-case-dominant regime.

The following illustrates the benefits of an $(n, n-1)$ pattern-eliminating code compared to that of the single-parity-check code. In particular, the results provide a deeper insight into the behavior of a system as a function of the channel signatures. Subsequent results also explore the effectiveness of standard channel conditioning methods in offseting the rate penalty of codes in bandwidth-limited regimes. Finally, the previous experimental work is also revisited and reinterpreted in the light of both the theoretical and the simulation results. However, the discussion opens with a few remarks regarding the methods of biasing the system parameters, commonly used when estimating low-probability error events, and its effect on the system's performance picture.

2.5.1 Biasing the System Parameters

Before discussing any simulation results, it is necessary to address the issue of biasing the performance of a given system for the purpose of performance estimation. Since it is impossible to observe the error rates of interest in high-speed links through Monte Carlo simulation, it is common practice to modify the system's parameters to observe the behavior of an "equivalent" system at a lower error rate and subsequently extrapolate the results. Typically, a measurable error rate is achieved through some combination of minimum distance, noise variance and channel length. Although such adjustments offer a convenient way of studying the system's behavior over a range of operating conditions, more than biasing the error probability alone, they also affect the dominant noise mechanisms.

For instance, consider the case of a high-speed link operating in the quasi-worst-case-dominant regime at an error rate of 10^{-10}. Relying on Monte Carlo simulation and subsequent extrapolation to determine whether some error-control code succeeds in reducing the error down to the 10^{-15} target requires biasing the system parameters. However, increasing the noise variance reduces the value of f and thus inhibits the improvement of any code optimized for this regime. On the other hand, shifting the decision threshold so that sufficiently many possible values cross over and thus increase the likelihood of an error, as is often done in practice for high-speed links,

can yield the behavior previously illustrated in Figure 2-6, where the system no longer operates in the quasi-worst-case-dominant regime due to a negative decision distance.

The lesson to be drawn from the above discussion is that, when biasing the performance of a given communication system, the degree of the bias depends on the target performance, but the choice of a biasing method depends on the underlying error mechanism. In practice, the parameter f can be evaluated analytically for any system—in fact, Chapter 3 suggests a practical numerical approach. Thus ensuring that the value of f remains within some target range while biasing the system parameters, one ensures that the biasing preserves the nature of the underlying error mechanisms. However, the noise variance and the distance to the decision threshold alone may not provide sufficient control. Thus, contrary to the practical intuition, it may be necessary to alter the channel response in order to make the simulated system more realistic.

2.5.2 Code Performance in the Quasi-worst-case-dominant Regime

The purpose of this section is two-fold. First, it illustrates the behavior of codes in the quasi-worst-case dominant regime. These observations allow for the subsequent reinterpretation of the previous experimental results in the light of the results developed in this thesis. Second, it explores possible channel-conditioning methods and their potential in offsetting the rate penalty inherent to channel coding over bandwidth-limited channels.

Channel Signatures and Code Performance

This section focuses on the effect of the channel signature on the performance of both the SPC and the PEC. In the light of the previous results on biasing and due to a relatively limited supply of link channel measurements endowed with given properties, this section uses an emulated high-speed channel. Note, however, that

the subsequent sections on channel conditioning employ channel responses based on actual measurements [17].

The sample high-speed link channel used, shown in Figure 2-8[top], corresponds to a modified ATCA 32" backplane with bottom-level routing (ATCA B32 [17]), operated at 9 Gbps and equalized through a 10-tap linear zero-forcing equalizer (LE-ZFE), previously discussed in Section 2.4.3. At this particular data rate, however, the equalized dispersion coefficients are still sufficiently strong to bring the signal over the decision threshold in the absence of noise and regardless of the secondary coefficients. The channel is thus further modified by reducing the magnitude of all dispersion coefficients past the location of the eighth symbol. In addition, to make the result more realistic, the last dispersion coefficient is reduced[21] by a factor of two. The reduced-magnitude coefficients are now on the order of those due to the channel reflections and are included in the secondary part of the channel response. The latter is also truncated for the ease of calculations. The principal part of the channel contains eight symbols, including the pre-cursors, while the full channel is 48 symbols long.

By examining the signature of the principal part of the channel, it is evident that the latter is fully correlated, in the sense of the pattern-correlation defined in Section 2.3.1. In order to illustrate the effect of the channel signature on the system performance, two more channels are created by *altering only the signs* of the coefficients in the principal part of the channel. The resulting channels are moderately correlated (Figure 2-8[middle]) and uncorrelated (Figure 2-8[bottom]). The corresponding worse-case patterns are shown in Figure 2-9. Note that, by applying the criterion of Theorem 7 to the corresponding channel signatures, it follows that an $(8, 7)$ pattern-eliminating code is effective for both the correlated and the fully correlated channels.

Simulation results are obtained at two different noise levels, $\sigma_1 = 2 \times 10^{-3}$ and $\sigma_2 = 5 \times 10^{-4}$, which are, as discussed in Chapter 1, realistic in a high-speed link. The corresponding symbol error rates for the uncoded system, computed analytically as

[21]Note that this reduces the benefit of a pattern-eliminating code.

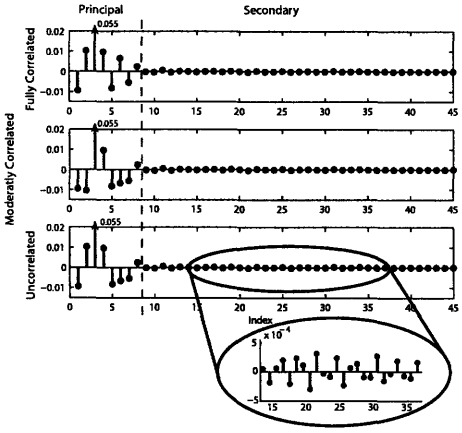

Figure 2-8: An Emulated High-speed Link – fully correlated channel[top], correlated channel[middle] and uncorrelated channel[bottom]

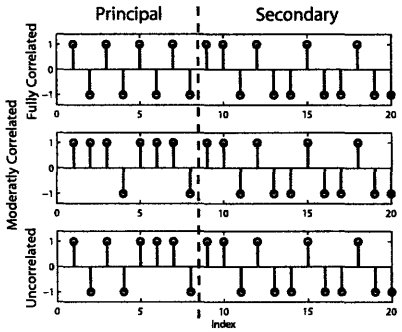

Figure 2-9: The Corresponding Worse-case Patterns – fully correlated channel[top], correlated channel[middle] and uncorrelated channel[bottom].

discussed in Chapter 3, are $p_{err1} = 6.9\times10^{-4}$ and $p_{err2} = 5.6\times10^{-5}$, respectively. Note that the channel signature has no effect on the symbol error rate for unconstrained transmissions. Considering both systems to operate in a quasi-worst-case-dominant scenario, the corresponding f values are, $1-f_1 = 2.6\times10^{-3}$ and $1-f_2 = 2.0\times10^{-13}$. Note that, since the system is assumed to operate in the quasi-worst-case-dominant regime, the quantities of interest, such as the correlation or the worst-case interference, are all referred to the principal part of the channel.

Tables 2.5.2 and 2.5.2 provide the Monte-Carlo simulation results for the $(7,6)$ single-parity check code (SPC) and the $(8,7)$ pattern-eliminating code (PEC). The quantity p_{err} *meas.* is the symbol error probability for the SPC prior to decoding, which, compared to the corresponding value of p_{err} for each table, indicates whether the SPC has any pattern-eliminating properties. The quantities *WER SPC BSC* and

	p_{err} meas.	WER SPC BSC	WER SPC meas.	PEC pred.	PEC meas.	% WC removed
Full Corr.	6.94×10^{-4}	1.0×10^{-5}	3.4×10^{-4}	1.8×10^{-6}	1.1×10^{-6}	100%
Med. Corr.	6.99×10^{-4}	1.0×10^{-5}	3.0×10^{-6}	1.8×10^{-6}	2.2×10^{-6}	100%
Uncorr.	6.86×10^{-4}	9.9×10^{-6}	2.48×10^{-7}	2.8×10^{-5}	3.0×10^{-5}	93.7%

Table 2.1: Code performances over three high-speed link channels with $\sigma_1 = 2 \times 10^{-3}$, $p_{err1} = 6.9 \times 10^{-4}$ and $1 - f_1 = 2.6 \times 10^{-3}$.

WER SPC meas. refer to the word error rates of the SPC as predicted assuming the errors to be independent and as measured in simulation. Specifically, assuming the ISI to be negligible, the standard expression is

$$\text{WER BSC } = 1 - (1 - p_{err})^n - n(1 - p_{err})^{n-1} p_{err} \tag{2.24}$$

where n is the corresponding codeword length. The quantity *PEC pred.* is the predicted symbol error rate of an $(8, 7)$ pattern-eliminating code computed by the result of Theorem 3 as

$$\text{PEC pred. } = 2(1 - f)p_{err}, \tag{2.25}$$

assuming that the pattern-eliminating code removes all occurrences of the worst-case interference. The quantity *PEC pred.* is the measured analogue. Finally, *% WC removed* denotes the percentage of removed worst-case interference events, where, as before, the worst-case interference refers to the primary part of the channel. Note that for the uncorrelated channel where the $(8, 7)$ PEC cannot remove all instances of the worst-case symbol pattern, *PEC pred.* is approximated by scaling down the error rate by the proportion of eliminated worst-case patterns, that is, PEC pred. $\approx$ $p_{err} \times$ (% WC removed)/100.

Sequentially examining each of the quantities simulated, the lack of substantial change for the measured symbol error rate, p_{err} *meas.*, as a function of the channel signature indicates that the SPC has no significant pattern-eliminating benefit over these channels. However, observing the discrepancies between the word error rates (WER) measured for the SPC to those predicted assuming the errors to occur independently further confirms the importance of channel signatures in quasi-worst-

	p_{err} meas.	WER SPC BSC	WER SPC meas.	PEC pred.	PEC meas.	% WC removed
Full Corr.	5.63×10^{-5}	6.7×10^{-8}	1.7×10^{-4}	1.1×10^{-17}	0 *	100%
Med. Corr.	5.55×10^{-5}	6.5×10^{-8}	0 †	1.1×10^{-17}	0 *	100%
Uncorr.	5.54×10^{-5}	6.4×10^{-8}	0 †	2.2×10^{-6}	1.9×10^{-6}	93.7%

Table 2.2: Code performances over three high-speed link channels with $\sigma_2 = 5 \times 10^{-4}$, $p_{err2} = 5.6 \times 10^{-5}$ and $1 - f_2 = 2.0 \times 10^{-13}$. Note that the " * " indicates estimates computed based on 10^9 symbols, while the " † " indicates those computed based on 10^{10} symbols.

case-dominant regime. As expected, for the fully correlated channel where multiple codeword errors become more likely, the observed WER is inferior to the predicted one. The discrepancy is roughly an order of magnitude at larger noise and two orders of magnitude at smaller noise. This is consistent with the fact that decreasing the noise variance increases the value of f and, with it, amplifies the importance of the worst-case symbol patterns and their correlation properties. Similarly, for the uncorrelated channel, where in the limit $f \rightarrow 1$ multiple errors occur with zero probability, the SPC performs better than predicted. The discrepancy is roughly two orders of magnitude at larger noise. At smaller noise, the confidence of the Monte Carlo estimate is not sufficient to allow for a meaningful comparison.

The simulated behavior of the pattern-eliminating codes yields the most interesting results. The most drastic of those is that, on channels where the $(8, 7)$ pattern-eliminating code is effective and noise is relatively small, the symbol error probability decreases by roughly *twelve orders of magnitude.* Since such error rates cannot be captured with 2×10^9 information symbols (discounting the constraint symbols), the Monte Carlo estimate records no errors. As expected, the improvement is significantly reduced at larger noise.

Finally, note that the observed proportion of worst-case patterns sucessfully eliminated by the PEC on the uncorrelated channel is the same regardless of the noise variance. This is due to the fact that the *effectiveness* of a pattern-eliminating code is independent of noise, but its *benefit* for a given system, in terms of the corresponding error-rate reduction, is not.

Code Performance With Rate Penalty

The previous section shows that a simple $(8,7)$ pattern-eliminating code can yield significant improvements in adequate channel conditions. However, as discussed in Section 2.4.3, due to the bandwidth-limited nature of the high-speed link, it is not sufficient to consider the resulting performance improvements compared to an uncoded system operating at the same data rate. Instead, the baseline for the comparisons is the uncoded system operating at the lower, *equivalent information rate* and equalized in a usual manner.

Following the guidelines of Section 2.4 for achieving a large f, it is not difficult to "fabricate" a communication channel for which an arbitrarily-large benefit can be derived. For instance, the channel of Fig. 2-10 is created based on a standard sinc function. Only half of the waveform is used, letting the peak value represent the principal signal component. All other interference coefficients are tapered by $\left(\frac{L-t-1}{5L}\right)^2$, where t is the time index[22], and the total response is normalized to unit L_1 norm. The error rates on this channel for a $(7,6)$ pattern-eliminating code are shown in Table 2.3 for zero-mean additive white Gaussian noise with $\sigma = 10^{-3}$.

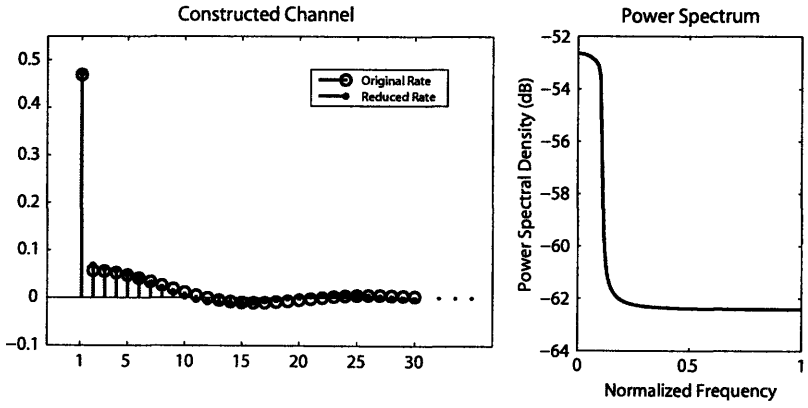

Figure 2-10: Sample Fabricated Channel – a) Pulse response, b) Power spectrum.

The results of Table 2.3 show that, for the above channel, a pattern-eliminating code drastically reduces the error rate *even with the rate penalty taken into account*, that is, comparing the coded performance to that of the uncoded system at the

[22]The interference coefficients need to be reduced in order to render the minimum decision distance non-negative.

(n,k) PEC	Uncoded p_{err} rate R	Coded p_{err} rate R	Uncoded p_{err} rate $\frac{kR}{n}$
$(7,6)$	2.6×10^{-9}	4.1×10^{-20}	5.1×10^{-9}
$(6,5)$	2.6×10^{-9}	0	5.3×10^{-9}

Table 2.3: Error Rates for a Fabricated Channel

same information rate. In particular, the $(7,6)$ code yields an error-rate reduction of eleven orders of magnitude, while the $(6,5)$ code prevents virtually all errors. Since the channel's frequency response of the fabricated channel is suggestive of standard bandwidth-limited systems, there is a potentially large benefit to exploring the application of pattern-eliminating codes to some common systems with ISI. This leads to the notion of channel conditioning, which mitigates certain channel characteristics, such as frequent nulls in the dominant portion of the channel, that may otherwise impair the performance of a pattern-eliminating code.

Channel Conditioning for Pattern-eliminating Codes

As demonstrated in Section 2.4, the performance of a pattern-eliminating code is directly tied to both the signature and the magnitude profile of the channel coefficients. The coding and equalization problems are therefore no longer separable. To quantify the effect of a particular equalizer on the code performance, consider the dispersive high-speed backplane link channel whose pulse and frequency response are shown in Figures 2-11 a) and b). Note that the channel is extracted from the M32 ATCA [17] measurements. In this example, the channel is equalized with a 2-tap finite-order impulse response filter. The filter coefficients are given by c_1 and c_2 where, to satisfy the peak power constraint at the transmitter, $|c_1| + |c_2| = 1$. The coefficient values are explored exhaustively and the corresponding error rates for the coded and uncoded system are shown in Figure 2-12. The three distinct plots correspond to three distinct codeword lengths displayed in the order of increasing overhead, that is, for $n = 7, 6$, and 5, respectively. All three scenarios operate at a coded rate of 9 Gbps and varying information rates. Note that, in practice, c_1 and c_2 are set either according to a zero-forcing criterion or an eye-maximization criterion [6, 9]. In each plot, the

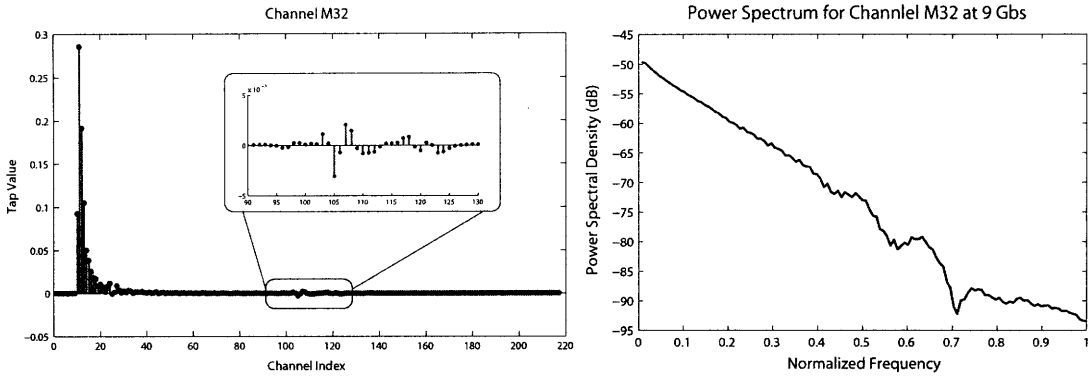

Figure 2-11: A High-speed Link Channel – Unequalized ATCA M32 Channel [17] operating at 9 Gbps: a) Pulse Response, b) Frequency Response.

intersection of the vertical lines with the error-rate curve marks the performance of the pattern-eliminating code for the two equalization criteria.

In all three figures, the pattern-eliminating code yields a significant benefit compared to the coded system operating at the same data rate. For $n = 7$ the symbol-error probability is reduced by four orders of magnitude, *fifteen* orders of magnitude with $n = 6$ and is "eliminated" with $n = 5$. Furthermore, inspecting the performance of the pattern-eliminating code in conjuction with an equalizer optimized according to either zero-forcing or eye-maximization criteria confirms that the equalization and coding problems are no longer separable. Specifically, neither criterion generally matches with the filter coefficients which optimize the performance of the pattern-eliminating code[23]. For instance, for $n = 6$, applying the standard zero-forcing criterion reduces the code benefit by over *ten* orders of magnitude.

The results of Figure 2-12 also indicate that the pattern-eliminating code provides no performance gain over an ATCA link channel when the rate penalty is taken into account. Precisely, the error-rate reduction achieved by decreasing the signalling rate of an uncoded system down to the equivalent information rate exceeds the error-rate reduction achieved with a pattern-eliminating code. Thus, contrary to the behavior observed for the "fabricated" channel, a pattern-eliminating code has little benefit over an ATCA link channel, when considered solely as an error-reduction technique.

[23]The fact that the eye-maximization is the optimal choice for the $(6, 5)$ pattern-eliminating code over this channel is accidental, as it is not optimal for the other two cases.

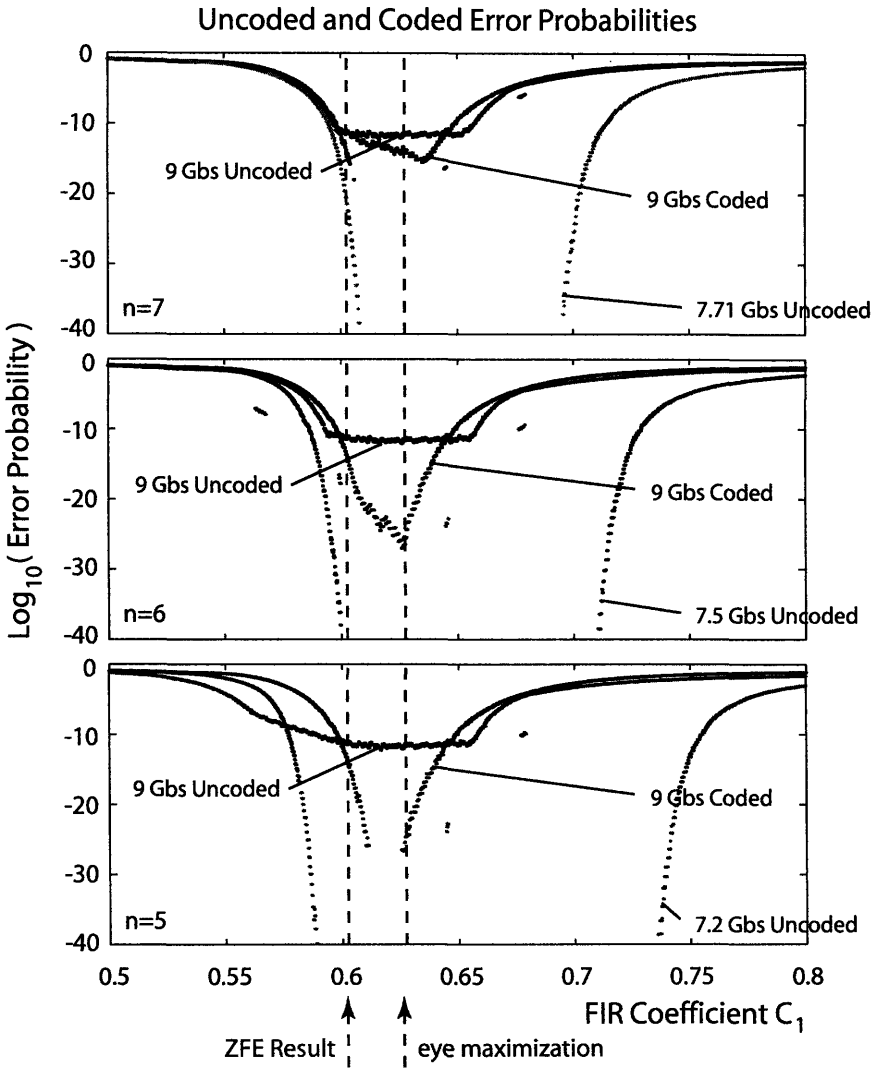

Figure 2-12: High-speed Link Performance – Each plot displays the error behavior, as a factor of the equalization type, for a coded system operating at 9 Gbps, an uncoded system operating at 9 Gbps, and an uncoded system operating at an equivalent information rate. The abscissa indicates different values of the filter coefficient $|c_1|$. The pattern pattern-eliminating code. The three plots correspond to three different pattern-eliminating codes, namely a) $(7,6)$, b) $(6,5)$, and c) $(5,4)$. The vertical lines correspond to filter coefficients for a zero-forcing equalizer and eye-maximization equalizer.

The discrepancy between the two behaviors is due to drastically steeper roll-off at the Nyquist frequency for the ATCA channel, compared to that of the fabricated channel.

The above example indicates that further work is required on channel conditioning techniques to reduce the effect of the rate penalty. However, the pattern-eliminating codes combined with the above simple jointly-optimized equalization technique yield significant improvements in systems that already incur some rate penalty. Specifically, in timing-sensitive systems run-lenght-limiting (RLL) codes are typically employed in order to force signal transitions required for timing-recovery circuits. Consider a system where it is necessary to eliminate all runlengths greater than some $l > 1$. By the result of Theorem 10, the $(n, n-1)$ pattern-eliminating code, with $n \leq l+1$, can replace a $(0, l)$-RLL code at a comparable or reduced rate penalty. Furthermore, if the $(n, n-1)$ code is effective in prohibiting all occurences of the worst-case patterns over that channel, the pattern-eliminating code also provides significant error-rate reductions at no additional overhead penalty. For instance, high-speed links commonly employ DC-balancing/run-length-limiting 8b/10b codes [18]. The latter are non-systematic block codes that bear no known pattern-eliminating properties in a high-speed link. Note that the 8b/10b and the (6,5) RLL-agumented pattern-eliminating code both allow for a maximum runlength of at most five symbols. However, the (6,5) pattern-eliminating code incurs less overhead. Furthermore, for the high-speed link channel of Figure 2-11, the (5,4) pattern-eliminating code reduces the error probability from 10^{-10} down to less than 10^{-40}. In addition, any $(n, n-1)$ pattern-eliminating code allows for simple encoding and requires virtually no decoding. However, the performance gain comes at the expense of the DC-balancing property also achieved by the 8b/10b code. Thus, the pattern-eliminating code is preferable to the 8b/10b code when the error, overhead and complexity reduction outweighs the benefits of DC-balancing.

The results of this section show that a suitable equalization technique can significantly improve the performance of a pattern-eliminating code. The resulting jointly-optimized equalization is, in principle, more power-efficient since the equalizer is no longer required to achieve very low error rates. Conversely, assuming a separability

between the equalization and coding problems leads to potentially large performance penalties. It is thus preferable to employ simple, jointly-equalized equalization techniques to their more complex but code-unaware counterparts.

2.5.3 Revisiting Previous Experimental Work

In light of the developments of this chapter, revisiting the results of the previous work on coding for high-speed links provides both the context and a potential justification for the experimentally-observed behaviors. As discussed in Section 2.1.3, the main body of work on this topic is a thesis by M. Lee [4]. Based on hardware implementations of different codes on a variety of high-speed links, Lee observes the following behaviors. For links operating at data rates between 5 Gbps and 6.75 Gbs and at error rates in the range of approximately 10^{-12} up to 10^{-3}, Lee notices that the errors occur almost exclusively in "clusters" but that, due to the low error rates, the clusters occur with low probability. Since the clusters are relatively large, that is, on the order of 10 errors occuring jointly, Lee correctly concludes that most error-correcting codes are impractical for high-speed links due to large overhead. She also correctly observes that error-control codes used for error detection are more overhead-efficient, and goes on to recommend implementing automatic-repeat-request (ARQ) for high-speed links, given the availability of a suitable feedback channel [7]. Finally, she observes that the burst-error-control codes perform poorly since the error spread is too large to be handled with practical overhead. However, Lee is uncertain as to the underlying reason for the large error correlation and thus assigns it, by default, to some correlated circuit noise inherent to the hardware setup.

Based on the results of Sections 2.2 through 2.4, the systems studied by Lee potentially operate in the quasi-worst-case-dominant regime. A first indication is the error dependence, evidenced in Figure 2-13 reproduced from [4]. Based on a measured average bit error rate, the plots compare the probabity of observing multiple-bit errors computed based on an independence assumption against the measured probabilities. In particular, the measured probability of multiple-bit errors is significantly larger than predicted. Although Lee attributes the resulting discrepancy to possible sources

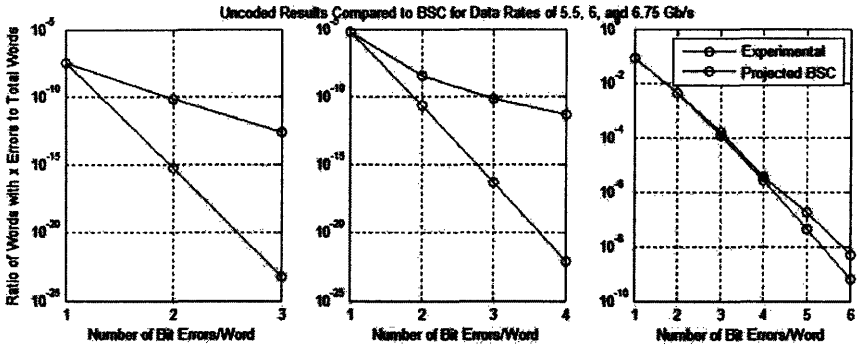

Figure 2-13: Figure 19 [4]: The three graphs show the behavior of the link at three different data rates: 5.5 Gb/s (left), 6 Gb/s (middle), and 6.75 Gb/s (right). As the data rate increases, the link becomes more and more like a BSC.

of correlated noise, the behavior is consistent with that of a correlated channel in the worst-case or quasi-worst-case-dominant scenarios.

Furthermore, since in the quasi-worst-case-dominant scenario, the probabilities for certain codeword symbols increase drastically when conditioned on the occurrence of the corresponding quasi-worst-case interference pattern[24], one expects the spatial distribution of errors to be skewed towards joint occurrences. This is consistent with Lee's observations, and the following is a direction quotation from [4]:

> At 6 Gb/s, 99.94% of all bit errors occur in codewords with error lengths of less than 10. Most of these, however, are single-bit errors, which technically have error lengths of one. For words with multiple errors, 75% of the bit errors are still within a length of nine. At 6.25 Gb/s, these numbers are 99.90% and 87.21%, respectively.

Lee's statement requires some additional clarification. The error length is defined as the number of bits spanned by the first and the last error in one codeword. The majority of the errors observed are single bit errors since, even when conditioned on the worst-case interference, the error occurs only for a sufficiently large noise event. Thus, multiple-bit error events are still relatively rare. However, the majority of the observed multiple-bit errors are contained within the length nine, which suggests the length of the principal part of the channel is on the order of 10 symbols (bits).

[24] i.e. the worst-case pattern formed by only the primary part of the channel.

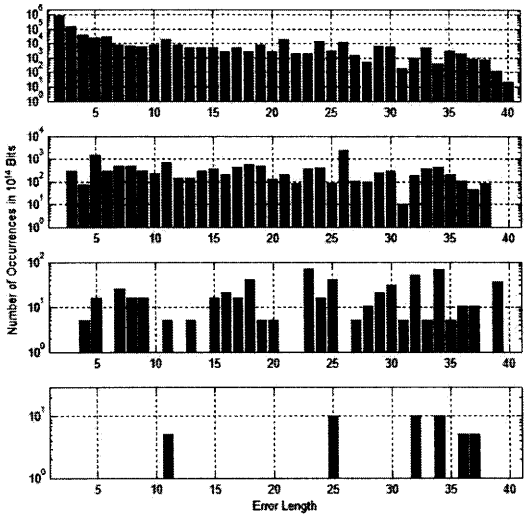

Figure 2-14: (Figure 21 [4]) Distribution of Error Lengths at 6.25 Gb/s on a Rogers Link. The error lengths of all 40-bit words with multiple errors is illustrated for a single test run, normalized to 1014 bits. The graphs indicate the error length distribution of error patterns of weight two (top), three (upper middle), four (lower middle), and five (bottom).

The conclusion that burst-error-correcting codes are inefficient for high-speed links is based on more detailed statistics regarding the distribution of error lengths in a codeword. These are reproduced in Figure 2-14. The plots deptict the error lengths measured for codewords of different *error weights*, where the error weight of a codeword is defined as the number of errors observed within that codeword. Note that the results for codewords of weight three or more, corresponding to the two lower plots, should be disregarded on account of the lack of statistical significance, brought about by the low probability of the underlying events. Assuming that the result corresponding to codewords of error weight two (the top plot) is statistically significant allows for several observations regarding the channel correlation and the underlying value of f. Specifically, in a codeword with exactly two symbol errors, an occurrence of two consecutive errors is at least an order of magnitude more frequent that the occurrence of two errors separated by one or more symbols. Employing the notation for the channel's pattern correlation, this suggests that $c\,(1, \mathbf{p}) = 1$. Furthermore, note that the distribution of errors corresponding to the subsequent error separations appears relatively uniform throughout the entire length of the codeword. For instance, the

number of codewords with an error length of five equals the number of codewords with an error length of twenty-one. This is suggests that the original hypothesis where the principal part of the channel spans approximately ten symbols, corresponding to the dispersion coefficients, may not be accurate. In particular, given the particularly low error rate, it is likely that there are approximately two to three significant channel coefficients. This type of channel response is due to three facts. First, Lee's system employs linear transmit and receive equalizers to reduce the channel dispersion, which can introduce nulls in the principal part of the channel. Moreover, due to the peak power constraint at the transmitter, linear equalization can amplify the interference coefficients in the secondary part of the channel, thus further reducing the value of f. Third, strong signal reflections, common in a high-speed link, add more weight to the tail of the channel compared to a typical bandwidth-limited channel. As a result, the value of f may not be as high as that observed for the ATCA channels described earlier. *It follows that, to apply a pattern-eliminating code to the channels studied by Lee, a more suitable form of channel conditioning is needed.*

The remaining significant result reported by Lee, that is, the "clustering" behavior of the error events, is an artifact of low error rates, and is thus vacuously consistent with the quasi-worst-case-dominant hypothesis. Specifically, shifting the focus from the error lengths to the number of errors in any given codeword, Lee considers blocks of uncoded symbols, each between 40 and 1000 consecutive symbols long, and observes the maximum number of errors, and the corresponding lengths, recorded in each block[25]. She observes that the maximum number of recorded errors remains the same, regardless of the block length. Even more precisely, the number of m-bit errors, where $m = 1, 2, \ldots$, remains constant with the codeword length. Lee's original results are reproduced in Figure 2-15. The reported behavior is principally due to the low probability of the observed events. Specifically, for an average error probability of 10^{-7}, registering a significant increase in the number of errors observed when increasing the block size from 40 bits to 1000 bits is very unlikely. Based on these

[25]Note that in order to avoid any boundary artifacts such as observing an error cluster split between two blocks, Lee adopts a sliding window approach. More precisely, for each block, Lee chooses the starting point in order to observe the maximum number of errors.

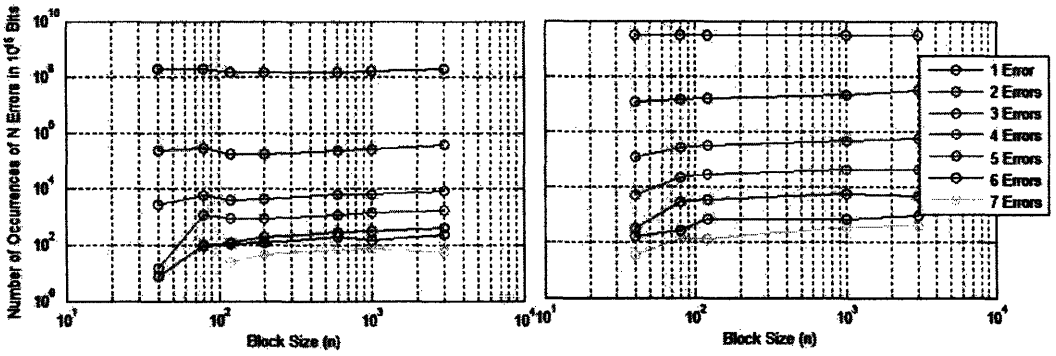

Figure 2-15: (Figure 20 [4]) Distribution of Errors Per Word at 6 Gb/s and 6.25 Gb/s: The graphs show the number of x bit errors per word over multiple block sizes that occurred for x between one and seven. The graph on the left is for 6 Gb/s, and the right is for 6.25 Gb/s. Each column of circles in the graph indicates a separate test run through the link. Although there are a few instances of words with more than seven errors, not enough occurrences were observed to accurately predict a trend. The y axis is normalized to 1015 bits. The results are actually based on 10^{14} to 10^{15} bits for 6 Gb/s and 1012 to 1013 bits for 6.25 Gb/s.

results, Lee correctly infers that error-correction codes can provide a benefit over the uncoded lower-rate system if sufficiently long codewords are used. This is due to the fact that, for a fixed error-correction capability, increasing the number of information bits in a codeword decreases the overhead. However, she also cautions that sufficiently long codewords may not be practical in high-speed links. Instead, she recommends error-detection codes, which require less overhead for the same error-detection capability.

Finally, Lee's results also provide an instance of the large-set-dominant behavior. More precisely, Lee observes that, as the data rate increases and the communication channel consequently degrades, the errors occurring on distinct symbols become effectively independent. She refers to this condition as having a binary symmetric channel (BSC) and writes:

> Figure 19 [corresponding to Figure 2-13 in this section] shows the probability of having x bit errors per word for reasonably small x based on experimental data for a wide range of data rates. This is compared with the projected probabilities for a BSC. For high data rates where the bit error rates are very low, the channel is similar to the binary symmetric

> channel model. Although it is not possible to determine why the link produces these results with 100% confidence, it is likely that reflections are the dominant noise sources, and the equalizers are not able to combat them. Reflections are the likely candidate, because the backplane is not counterbored, and reflections are generally uncorrelated.

Lee's justification is consistent with the behavior discussed in Section 2.3.2. Specifically, as the data rates increase, both the dispersion coefficients and the signal reflections increase. Due to the limited equalization and the use of linear equalizers, the residual dispersion and reflections combine to yield a channel response that is is relatively uniform over a potentially large number of symbols. When the residual interference is sufficiently large render the minimum decision distance[26] sufficiently negative, the quasi-worst-case-dominant scenario no longer applies. The stronly negative minimum distance is consistent with the dramatic increase in the error rate recorded in Lee's experiment. More precisely, the result of Figure 2-13 indicates that the error rates at which the error events can be considered independent are on the order of 10^{-1}. Note that Figure 2-13 also provides an example of a system where the operating regime changes from large-set-dominant to quasi-worst-case-dominant as the error rates decrease. This serves to further illustrate that, when either biasing the system's operating parameters or extrapolating the behaviors observed at higher error rates to behaviors predicted at lower error rates, it is necessary to consider the operating regime throughout the entire range of conditions. An analytical method of achieving this is described in the following chapter.

In addition to Lee's thesis, two smaller-scale studies, [11] and [12], also explore the possibility of error-control coding for high-speed links. In both, error-control codes were found to yield several orders of magnitude of improvement under certain conditions. Based on the magnitude of improvement alone, it is in principle difficult to determine whether the benefit is due to

- a link operating in large-set conditions discussed in Section 2.3.2, which render classical error control codes suitable;

[26] Relative to the principal part of the channel.

- a link operating in a quasi-worst-case-dominant regime, where the principal part of the channel is weakly correlated;

- a link operating in a quasi-worst-case-dominant regime, where the principal part of the channel is correlated and due to some incidental pattern-eliminating property of the error-control code.

The last interpretation is particularly tempting in [12] where the authors implement a hybrid MTR/ECC[27], suggestive of distance-enhancing MTR codes for partial response channels. However, the detailed analysis, also provided in [12], of the corresponding test results provides a deeper insight into the system's operating conditions. More precisely, the authors consider two high-speed link channels, one corresponing to a 10" backplane and another to a 40" backplane. For the 10" backplane, the authors observe that the error-control code may provide a significant benefit over the uncoded case at the reduced data rate. The uncertainty is due to the low error rate and the fact that the improvement cannot be measured by Monte Carlo trials. The improvement is thus predicted analytically, assuming the errors to occur independently. However, as the authors correctly observe, the principal limitation of this result is the fact that the uncoded system, which is also unequalized, already reches the target error rates of 10^{-12}. Thus, there is no practical benefit to implementing an error-control code.

Still concerning the results of [12], a more interesting result is obtained considering the 40" backplane. The corresponding Monte Carlo trials are reported to yield "inconsistent" results. Specifically, it is observed that some PRBS[28] sequences yield no errors while others yield relatively high error rates. In the light of the previous discussions, the behavior is likely due to the system operating in the quasi-worst-case-dominant regime. From the physical perspective, increasing the backplane length exacerbates different types of material and dielectric losses and consequently increases the signal dispersion. From the perspective of the reported test results, the number of occurrences of the corresponding worst-case patterns in the PRBS sequence strongly determines the outcome of the corresponding Monte Carlo trials. In this case, given

[27]MTR –Maximum-transition-run, ECC –Error-correcting code.
[28]Pseudo-random bit stream.

that the corresponding PRBS has length 2^{11}-1 symbols and that the channel dispersion[29] is likely on the order of 10 symbols, it is possible for different PRBS patterns to yield significantly different statistics for the occurrence of the worst-case symbol patterns.

The recurring lesson from these and previous results is that for ISI-and-AWGN-limited channels, both the practical intuition and the research methodology often need to be revised depending on the operating regime.

2.6 Summary

In the large-noise and large-set-dominant regimes, classical coding theory provides an exhaustive characterization of different error-control codes. The remaining issue of hardware complexity has already been partially addressed in [4]. Instead, this chapter focuses on coding techniques for worst-case-dominant or quasi-worst-case-dominant regime. It demonstrates that, in these regimes, channels with weak correlation[30] impose more lenient requirements on standard error correction codes. For instance, a single parity check code corrects all codeword errors as long as the codeword length is strictly lesser than the channel's minimum correlation length. However, standard error correction codes implemented over correlated channels, such as the dispersive channel, incur a potentially large overhead, due to the fact that the principal part of the channel response in a high-speed link is on the order of ten symbols or less.

Instead, this chapter develops an alternative approach to coding in the worst-case-dominant or quasi-worst-case-dominant regime. In particular, it develops the pattern-eliminating codes, which provide a performance improvement by preventing worst-case interference events. The $(n, n-1)$ pattern-eliminating codes allow for simple encoding and require virtually no decoding. They are effective for many channels of interest, including the dispersive channel. The simulation results show that an $(n, n-1)$ pattern-eliminating code can provide error-rate reductions of over fifteen

[29] And, thus, the length of the principal part of the channel, which causing the worst-case-dominant behavior.

[30] In the sense of pattern-correlation

orders of magnitude on realistic high-speed link channels. However, such benefits often vanish when the rate penalty of a code is taken into account. A method of overcoming the rate penalty of a pattern-eliminating code is to endow the code with additional timing benefits. In particular, for most channel signatures, the $(n, n-1)$ pattern-eliminating code can be augmented by a simple rule to systematically prohibit all runlengths of more than $n-1$ ones or negative ones[31].

While the $(n, n-1)$ pattern-eliminating code is effective over many channels of interest, the $(n, n-2)$ code is effective over any channel. However, when both codes are effective, the $(n, n-1)$ code is preferable due to lower overhead and encoder simplicity. In general, the benefit of an $(n, n-c)$ pattern-eliminating code for $c > 2$ is the potential of providing a larger separation from the worst-case patterns. However, given that the principal part of the high-speed link channel is typically on the order of ten symbols or less, the rate penalty incurred by letting $c > 2$ likely exceeds the benefit.

Further work is required on extending the pattern-eliminating properties to deal with a wider range of operating conditions. In particular, one of the remaining problems consists of identifying or developing suitable equalization or channel conditioning techniques that optimize the performance of a pattern-eliminating code. Such equalization is, in principle, significantly more power-efficient compared to that employed in current high-speed links, as the equalizer is no longer is the sole method of achieving the required low error probability. The corresponding scheme could potentially yield significant benefits for high-speed links by enabling the communication at higher data rates than those achieved previously, or by providing the same signalling speeds at a greater energy efficiency.

[31] That is, provide the capability of a $(0, n-1)$-RLL code.

Chapter 3

Performance Estimation of Coded High-speed Links

The task of simulating a coded high-speed link is plagued with difficulties. The low error rates, typically on the order of 10^{-15}, render Monte Carlo simulation strictly infeasible, even at today's computing speeds. Yet, as discussed in Section 2.5.1 of the previous chapter, biasing the system's operating parameters to inflate the frequency of the error events may change the corresponding interference-and-noise regime and therefore fundamentally change the system's behavior in response to a code. For instance, an increase in the noise variance steers a system away from a (quasi-)worst-case-dominant regime and therefore renders a pattern-eliminating code ineffective. Furthermore, the theoretical expressions developed in the previous chapter principally concern the limiting cases. Although a well-equalized high-speed link is likely to operate in the limit of a large-set-dominant scenario, and similarly for a link with low noise and large dispersion operating in the quasi-worst-case-dominant scenario, the limiting error expressions computed in the previous chapter may not be sufficiently accurate for the in-between cases, especially at the low error probabilities.

This chapter introduces convenient methods of estimating the performance of a coded high-speed link under different regimes. It also introduces an efficient computational algorithm for accurately quantifying the effect of a code on symbol error probabilities regardless of the operating regime. The latter is also of use for practi-

cioners, as it can be readily incorporated into the existing high-speed link simulation software. The discussion begins with an overview of the previous work, followed by the development of different simulation methodologies and a description of the new algorithm.

3.1 Preliminaries

This section revisits the abstracted model of a coded high-speed link from the perspective of performance estimation and reviews the relevant previous work. Note that, although the main body of previous work on coding for high-speed links is predominantly experimental due to the former lack of theoretical and analytical framework, the issue of *uncoded* performance estimation for high-speed links has been extensively addressed.

3.1.1 System Model

The underlying communication system assumed in the upcoming development is the equivalent discrete-time coded system of Section 2.1.2 of the previous chapter. As a reminder, the received signal at some time instant $i \in \mathbb{Z}$, denoted by $Y_i \in \mathbb{R}$, is represented as $Y_i = Z_i + N_i$, where Z_i is the signal component due to the transmitted symbol and the inter-symbol interference (ISI) and N_i is the zero-mean additive white Gaussian noise component of variance σ^2. Again, the noise is assumed to be independent of the signal at all times. Based on some channel response of length L given by coefficients $h_0, \ldots, h_{L-1}$, the noiseless signal component Z_i is written as

$$Z_i = \sum_{k=0}^{L-1} X_{i-k} h_k \tag{3.1}$$

where, for convenience of representation, it is assumed that the channel has no pre-cursor coefficients. However, all the results presented in this chapter generalize readily to the pre-cursor case.

Since the noise distribution is known, this chapter focuses on statistics regarding Z_i. Let $f_{Z_i|X_i}$, denote the marginal probability mass function associated with the random variable Z_i, conditioned on the symbol transmitted at time i. Similarly, let f_{N_i} and $f_{Y_i|X_i}$ denote the marginal densities for the random variables N_i and Y_i respectively. Since the noise is assumed to be statistically independent from the signal at any time, it follows that

$$f_{Y_i|X_i} = f_{Z_i|X_i} * f_{N_i} \tag{3.2}$$

where the $*$ operator denotes the convolution[1].

The probability distribution $f_{Z_i|X_i}$ can be interpreted as a decision distance profile. For a system employing soft decision decoding, the corresponding $f_{Y_i|X_i}$ yields the marginal distribution for the soft information. For a system employing hard-decision decoding, integrating $f_{Y_i|X_i}$ over the error region yields the probability of incorrectly detecting the corresponding transmitted symbol X_i, prior to any error correction. Specifically, assuming the blind MAP detection rule discussed in Section 1.2, the unilateral error probability p_{err} is written as

$$p_{err} = P(Y_i < 0|X_i = 1) = \int_{-\infty}^{0} f_{Y_i|X_i=1}(y) \tag{3.3}$$

Note that for an unconstrained stream of equiprobable symbols, $f_{Z_i|X_i}$ is symmetric with respect to the realization of X_i.

Due to the channel memory, for any two time indices $i, j \in \mathbb{Z}$, $i < j < i - L$, signals Y_i and Y_j are *statistically dependent.* Thus, generally,

$$f_{Y_iY_j}(y_1, y_2) \neq f_{Y_i}(y_1) f_{Y_j}(y_2)$$

when $i < j < i - L$, and similarly for the distributions conditioned on X_i. It follows that for some (n, k) linear block code, the joint distribution of detection errors on the n codeword symbols cannot typically be obtained from the marginal densities

[1]Note that in order to represent the corresponding expression as a standard convolution, the discrete random variable Z_i needs to be represented as an equivalent continuous random variable. However, this abuse of notation is widely tolerated.

alone. Section 3.3 discusses the conditions where the independence holds and proposes alternative approaches to performance evaluation when the marginal probabilities are not sufficient.

3.2 Previous Work

As previously discussed, accurately simulating coded systems over channels with ISI is a task of varying difficulty. For systems operating at low error rates, Monte Carlo simulation is impractical due to the large sample size requirements. Sample-size reduction methods, such as importance sampling, exist, but long channel memory reduces their effectiveness due to the dimensionality effect [48]. Estimating the performance of such systems in practice principally relies on accurate analytical computations of signal probability distributions. Given that the residual ISI is widely acknowledged as an important error factor, especially at low error rates, several analytical tools for estimating the system's performance focus on computing probability distributions for Z_i alone. In [49] and [50], to calculate $f_{Y_i|X_i}$, the authors consider individual distributions associated with each transmitted symbol. More precisely, consider the transmitted symbol X_i and the corresponding received signal Y_i. From the Equation 3.1, it follows that the contribution to Y_i due to some symbol X_j, where $j = i, \ldots, i-L+1$, is given by $-h_{i-j}$ when $X_j = -1$ and h_{i-j} when $X_j = 1$. Assuming the transmitted symbols to be independent, the distribution of the overall received signal Y_i becomes the convolution of the densities corresponding to individual symbol contributions. Similarly, [51] operates on the same principle, except that the individual probability distributions are computed for 4 symbols at a time. However, expressing the probability density of the total received signal as the convolution of probability densities due to each symbol holds only for *independent symbols*, and does therefore not apply for coded transmissions in general. In addition, the symbol error probabilities constitute only the marginal statistics – information on joint statistics is needed to evaluate the performance of a coded system. In [12], it is assumed that the errors occur independently in order to capture the behavior of the system at low

error rates. However, the results of Section 2.5 demonstrate that a high-speed link can operate at a low error rate in the quasi-worst-case-dominant regime where errors events are highly correlated. Finally, note that due in part to the lack of adequate simulation framework. the main body of work on coding for high-speed links [4], as well as [11] and to an extent [12], remains exclusively experimental.

3.3 Performance Estimation Methodology

Revisiting possible noise and interference scenarios defined Chapter 1, the *large-noise scenario* offers no challenges: the error probability for any time index i is given by

$$p_{err} = P(N_i < -h_0)$$

and the errors are statistically independent for any finite set of distinct time indices. More precisely, since the noise is modeled as additive and white, the ISI is the only link between the error events occuring at distinct symbols. Therefore, in the limit where the ISI vanishes compared to the noise variance, so does the information that one received signal provides on another, and the errors events on received symbols also become independent. Section 2.2.2 of the previous chapter provides a convenient expression for evaluating the effect of ISI on the practical accuracy of the large-noise assumption.

Similarly, Section 2.3.2 of the previous chapter shows that, in the limit of the *large-set-dominant scenario*, errors on distinct symbols become effectively independent. Furthermore, from the perspective of high-speed links, the corresponding results suggest that the independence assumption is relatively accurate for well-equalized channels. This is illustrated in Section 3.5 at error rates that can be captured in Monte Carlo simulations. Regarding the effect of code constraints on the individual symbol error probabilities, Section 3.4 describes an efficient algorithm that accurately computes the marginal probability distributions for the random variable Z_i given any systematic linear block code. Note that the corresponding algorithm makes no as-

sumptions as to the nature of the noise and interference, and is therefore applicable to any scenario.

However, in the *worst-case-dominant* scenario, the joint error statistics and their effect on the behavior of error-control codes differs significantly from the intuition developed on well-equalized AWGN channels on which such codes are predominantly used. The following describes the performance estimation methods suitable in this case. A subsequent section also discusses the use of marginal probability distributions apart from the context of joint error statistics. However, since establishing the validity of either of the three scenarios requires comparing the a priori and a posteriori probability distributions for the random variable Z_i, the issue of computing these distributions is addressed first.

3.3.1 Computing the A Posteriori Signal Distributions

In the context of Chapter 1, the a posteriori signal distribution refers to the probability distribution of the random variable Z_i conditioned on the value of the corresponding transmitted symbol X_i and on the occurrence of the detection error. More precisely, following the convention of the previous chapters, it is assumed that $X_i = 1$ and conditioning on the error event is thus equivalent to conditioning on the event $Y_i < 0$. The corresponding conditional signal distribution is denoted by $f_{Z_i|Y_i<0,X_i=1}$.

The following expression, obtained by reformulating the conditional probability in the now-familiar fashion, yields the most convenient method of computing the conditional signal distribution:

$$f_{Z_i|Y_i<0,X_i=1}(z) = \frac{f_{Z_i|X_i=1}(z)\mathrm{P}(N_i < -z)}{p_{err}} \tag{3.4}$$

where, as previously,

$$p_{err} = \sum_{z\in\Omega} f_{Z_i|X_i=1}(z)\mathrm{P}(N_i < -z) \tag{3.5}$$

and Ω denotes the sample space of the random variable Z_i conditioned on $X_i = 1$. In that manner, the worst-case parameter f is computed as the a posteriori probability

of the worst-case interference z_{wc}, that is,

$$f = \frac{f_{Z_i|X_i=1}(z_{wc})\mathrm{P}(N_i < -z_{wc})}{p_{err}} \tag{3.6}$$

Similarly, for the quasi-worst-case-dominant scenario, denote by $\tilde{Z}_i$ the signal formed by only the principal channel coefficients and $\hat{Z}_i$ that formed by the secondary channel coefficients. Then the quasi-worst-case analogue of the worst-case parameter f, denoted by $\tilde{f}$ for convenience, is given by

$$\tilde{f} = \frac{f_{\tilde{Z}_i|X_i=1}(\tilde{z}_{wc}) \sum_{z\in\hat{\Omega}} f_{\hat{Z}_i|X_i=1}(\hat{z})\mathrm{P}(N_i < -\tilde{z}_{wc} - \hat{z})}{p_{err}} \tag{3.7}$$

where $\hat{\Omega}$ denotes the set of possible interference values caused by the secondary part of the channel alone and the symbol error probability p_{err} is computed based on the entire channel response. As a reminder for a channel of length L with the primary channel of length $\tilde{L}$, the quantity $\tilde{f}$ represents the a posteriori probability that the most recently transmitted symbol and the $\tilde{L}-1$ immediately preceding symbols form a worst-case pattern given that a detection error has occured. Note that the notion of the worst-case pattern in the quasi-worst-case-dominant scenario refers to the primary part of the channel alone.

In Equations 3.4-3.7, the quantity $\mathrm{P}(N_i < -z)$ for any $z \in \mathbb{R}$ is obtained by evaluating the cumulative distribution function of a Gaussian random variable, for which efficient numerical implementations exist [52]. The remaining challenge consists of determining the probability mass function $f_{Z_i|X_i}$. For an uncoded system, $f_{Z_i|X_i}$ can be computed efficiently as described in Section 3.2 even for relatively long channels. The generalized version of the algorithm for coded systems is described in Section 3.4.

3.3.2 Performance Estimation in the Worst-case-dominant Regime

In the context of performance estimation, the scope of the worst-case or quasi-worst-case-dominant scenarios is of practical nature. For the pattern-eliminating codes

(PEC), the suitable range of f, or $\tilde{f}$, corresponds to the region where the PEC yields a significant benefit, while for the classical error control codes (ECC), the corresponding range is set by computational constraints. The following two sections discuss both cases more thoroughly.

Performance Estimation of Pattern-eliminating Codes

Estimating the performance of pattern-eliminating codes in the worst-case or the quasi-worst-case-dominant scenarios is computationally straightforward. From Section 2.4.2 of the previous chapter, it follows that the error probability p'_{err} of a system implementing a pattern-eliminating code is bounded by

$$p'_{err} \leq 2^k(1-f)p_{err}$$

for the worst-case-dominant scenario, or

$$p'_{err} \leq 2^k(1-\tilde{f})p_{err}$$

for the quasi-worst-case-dominant scenario, where the parameters f or $\tilde{f}$ are computed as described in the previous section. Since, as $f \to 1$ or $\tilde{f} \to 1$ the respective bounds become tight, the above expressions are sufficient to accurately estimate the performance of a pattern-eliminating code at small f (or $\tilde{f}$), that is, in operating conditions where such a code provides a significant benefit.

Performance Estimation of Error Control Codes

Estimating the performance of error-correction codes requires knowledge of the joint error statistics. Since the experimental work on high-speed links focuses on linear block codes and since, based on the results of the previous chapter, error control codes in general[2] yield a limited benefit in (quasi-)worst-case-dominant regimes, the following discussion focuses on linear block codes alone.

[2]With the exception of the single parity check code, as also previously demonstrated.

For a system implementing a linear block code and operating in the worst-case-dominant regime, it suffices to identify the dominant error-causing symbol patterns and evaluate the joint error probabilities conditioned on these patterns. The number of patterns to consider depends on the value of f, computed beforehand as previously described. Thus, the question of classifying a system as worst-case-dominant for the purpose of performance estimation of linear block codes is a matter of computational complexity.

When f is sufficiently high so that the contribution of non-worst-case patterns to the error expression is negligible, the performance estimate becomes particularly simple. Consider some (n, k) linear block code and the corresponding codeword symbols $X_n, \ldots, X_1$. The corresponding received symbols $Y_n, \ldots, Y_1$ are function of the i.i.d.[3] noise random variables $N_n, \ldots, N_1$ and the $L + n - 1$ transmitted symbols $X_n, \ldots, X_{-L+2}$. Denote by S the set of all patterns $X_n, \ldots, X_{-L+2}$, taking into account the code constraints. Furthermore, define the function $g_N : \mathbb{N} \to \mathbb{N}$ so that, given a positive integer m representing the number of errors in a codeword, the output is the number of distinct $(n + L - 1)$-symbol patterns of the form $\mathbf{x} \in \mathrm{S}$, within which there are *exactly* m worst-case patterns nested. Note that, for each m, the corresponding $g_N(m)$ is obtainable from the pattern-correlation function introduced in the previous chapter. It suffices to observe the shifts at which the pattern correlation function has unit value and account for the multiplicity of the remaining "free" symbols. Then, the coded system's word error rate (WER) can be bounded by

$$p_{\mathrm{WER}} \leq \sum_{m=t+1}^{n} \frac{g_N(m)}{|\mathrm{S}|} \left(\mathrm{P}(N < -z_{wc})\right)^m \tag{3.8}$$

where $z_{wc} > 0$ is the minimum distance to the decision threshold due to worst-case interference and t is the error-correcting power of the code. The bound becomes tight in the limit $f \to 1$. In the quasi-worst-case-dominant scenario, the channel-related quantities are referred to the primary part of the channel alone. The bound thus becomes an approximation due to the contribution of the coefficients in the

[3]Independent and identically distributed.

secondary part of the channel. However, the approximation also becomes exact in the limit.

3.3.3 Further Use of Marginal Probability Distributions

While the previous section focused on the joint error statistics, it should be noted that the corresponding marginal probability distributions are useful in their own right. Most immediately, they provide more accurate information about system's margins, such as the signal swing[4]. Also, they provide accurate cross-over probabilities[5] for any symbol location in a codeword. This is particularly useful in the case where, for each codeword, some part of the information payload is more sensitive to error, like control or addressing information for instance. Finally, marginal probability distributions for the received signal may also be useful in soft-decoding of Reed-Solomon codes.

3.4 Efficient Computation of Probability Distributions for Coded Systems

The present section describes an efficient numerical algorithm that enables accurate computation of the probability distribution associated with the random variable Z_i given some systematic (n, k) linear block code. For the ease of representation, the following algorithm description relies on the simplified scenario illustrated in Figure 3-1, where both the length of the channel response and the time index i of the most-recently transmitted symbol are fixed relative to a given codeword. The subsequent development extends the corresponding results to arbitrary times indices, channel lengths, sources of interference and even symbol alphabets. Also note that since the signal is independent of the noise and the corresponding probability distributions are therefore decoupled, the underlying system is assumed to be noiseless. Thus, $Y_i = Z_i$.

[4] i.e. minimum and maximum signal values.

[5] In a system implementing a hard decoding scheme, integrating the marginal probability distribution for the received signal, conditioned on $X_i = x$, over the error region yields the probability that the symbol X_i is decoded incorrectly assuming an x was transmitted. That probability is referred to as the *cross-over probability* for the i^{th} symbol in the codeword and denoted by $p_{|}x^{(i)}$.

Given the two assumptions, it follows that the probability distribution of the random variable $Y_i|_{X_i=1}$ is entirely determined by the probability distribution of the random variable W_i, denoting the *contribution to Y_i from the preceding codeword.* Since

$$Y_i|_{X_i=1} = h_0 + W_i$$

and the contents of two distinct codewords are statistically independent, the probability distribution of $Y_i|_{X_i=1}$ will follow from the probability distribution of W_i by a simple transformation.

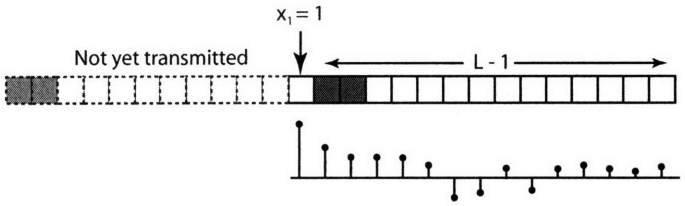

Figure 3-1: The simplified channel/code scenario – For a given (n, k) systematic linear block code, it is assumed that the most-recently transmitted symbol corresponds to the first symbol in the codeword. In addition, it is assumed that the previously-transmitted codeword is the sole source of interference. Both assumptions are revisited in Section 3.4.2.

3.4.1 Algorithm Description

When the set of allowed symbol sequences is restricted, straightforward enumeration provides a simple approach to calculating the probability mass function for the probability mass function f_{W_i} of the random variable W_i defined previously. Possible values of W_i are obtained by considering each element of the codebook individually. An (n, k) code allows for 2^k possible codewords, each n symbols long. Translating the resulting n-long possible symbol patterns into signal values using Equation 2.1 and noting that the patterns are equally likely yields a probability distribution. The size of the resulting probability mass function can be reduced by quantizing the possible signal values to some precision Δ and adequately grouping the corresponding probabilities. This mechanism is illustrated in Figure 3-2 for a simple (3,2) single-parity-check code with $\Delta = 0.1$. While the above method is adequate for dealing with

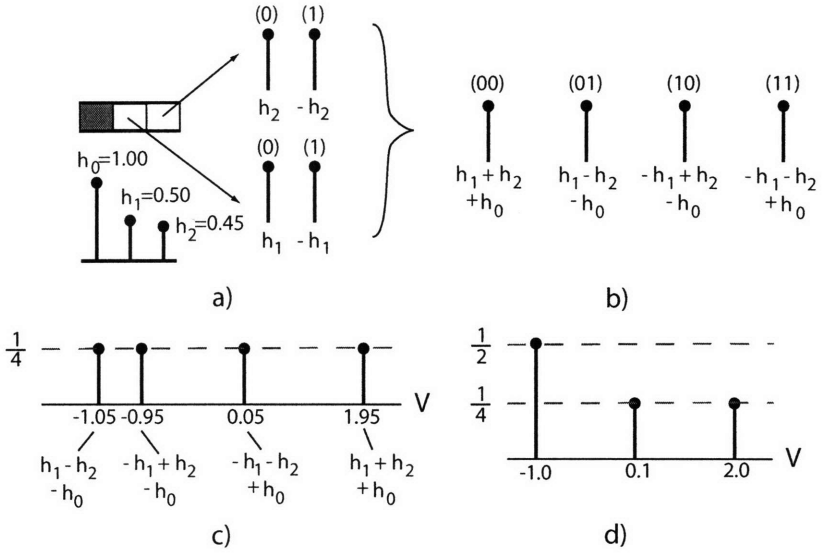

Figure 3-2: Example: Signal contribution due to one previously transmitted codeword – a) The corresponding portion of the channel response. b) The allowed symbol patterns. c) The resulting PMF. d) The quantized PMF.

codewords of relatively short to moderate lengths, it quickly becomes burdensome as the number of independent bits in a codeword increases. For instance, computing the probabilities associated with a codeword of $k = 100$ information bits and any number m of dependent bits over a realistically long channel[6] requires considering $2^{100} \approx 10^{30}$ distinct possibilities. In order to handle arbitrarily long codewords, it is therefore necessary to reduce the enumerative burden of the method.

Building on the naïve enumerative approach described previously, this section describes an efficient algorithm for computing the marginal probability distribution for the received signal contribution W_i given some code and effective channel response. The method of alleviating the computational burden of dealing with symbol patterns from a restricted set relies on two properties of a systematic linear block code. First, the fact that the information appears explicitly in the codeword yields a string of k unconstrained symbols which can be separated into independent, non-overlapping blocks. Second, the fact that the code is linear implies that the effects on the parity of each individual block added together entirely determine the parity associated with a particular information pattern. These two properties combined enable a reduction

[6]The results of Section 3.5 were obtained on a backplane channel of comparable length

in computational complexity.

A more computationally-oriented description of the above idea relies on the concept of *subcodeword*. A subcodeword is obtained by segmenting k unconstrained information symbols into independent, non-overlapping blocks and appending m parity symbols to each block. The code associated with a given subcodeword is the original code shortened to include only the relevant symbols. This is illustrated in Figure 3-3 for a (7,4) Hamming code. For ease of representation and without loss of generality, the subsequent development assumes that the information is subdivided into even, consecutive blocks of $d < k$ symbols.

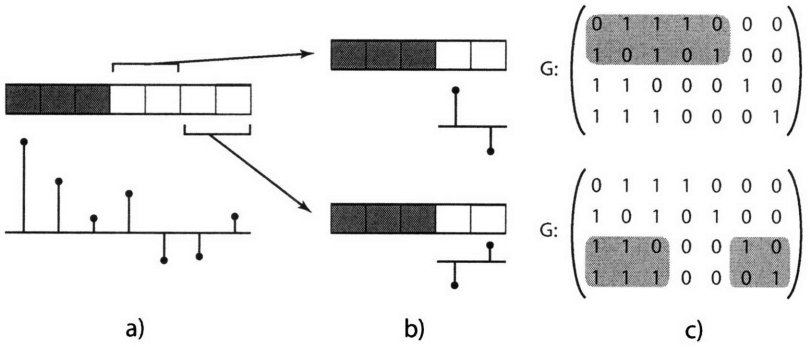

Figure 3-3: Example: Partitioning a (7,4) Hamming codeword for $d = 2$ – a) The original codeword. b) Two subcodewords. c) The portions of the generator matrix associated with each subcodeword.

The algorithm now reduces to the following three steps.

1. *Computing* the partial information for each subcodeword efficiently.

2. *Grouping* partial information to reduce storage requirements.

3. *Combining* partial information from each subcodeword.

The principle which allows for a reduction in the computational complexity is the computing and grouping of partial information at the *level of a subcodeword*. On the other hand, the method to efficiently combine the partial information to form the information due to a full codeword, identical to that obtained through a previously-described naïve approach, is due to a judicious choice of *partial information*. More precisely, defining the relevant subcodeword information as

- the contribution of the d information bits to the received signal;
- the contribution of the m parity bits to the full codeword parity.

allows for reductions at the level of the subcodeword without any loss of information at the level of the final probability distribution. The following sections individually describe each step of the algorithm.

Computing Partial Subcodeword Information

Since each subcodeword contains d unconstrained information bits, where d can be chosen conveniently small, proceeding by enumeration is no longer impractical. For each of the resulting 2^d possible subcodewords, the signal/parity contributions are represented by a scalar $w \in \mathbb{R}$ and an m-dimensional vector $\mathbf{b} = (b_1, b_2, ..., b_m)^T \in \{-1, 1\}^m$, where b_i is the subcodeword's contribution to the i^{th} parity bit and w its contribution to the received signal.

The quantities $\mathbf{b}$ and w can be computed in a "batch", by enumerating the possible 2^d bit patterns first. For this purpose, it is convenient to represent the corresponding patterns as two matrices, one corresponding to the symbols from the set $\{-1, 1\}$ and the other corresponding to the underlying bits, taking values from the set $\{0, 1\}$. The resulting value of w is obtained by multiplying the first matrix by a real vector $\mathbf{h}$ representing the corresponding channel coefficients. Similarly the vector $\mathbf{b}$ is obtained by multiplying the second matrix by generator vectors corresponding to each parity bit. Alternatively the same quantities can also be computed "sequentially", focusing on one information symbol at a time and considering the effect of its two possible values on both w and $\mathbf{b}$.

Grouping

Still operating at the level of a particular subcodeword, it can happen that two or more information patterns yield the same contribution to the received signal. This is particularly the case when signal contributions are quantized to some adequate

accuracy. In that case, there is often a large benefit to grouping the resulting signal/parity vectors with identical signal contributions. As an illustration, when the portion of the channel response which spans a particular codeword is approximately uniform (coefficients of equal magnitude), there will be roughly twice as many possible signal values as there are independent symbols. However, the total number of symbol patterns resulting from any subcodeword is exponenetial in the number of independent symbols. There will therefore typically be many more symbol patterns, and therefore the signal/parity vectors, than possible distinct signal values. As an illustration, consider a channel with coefficients of magnitude δ and a subcodeword with d information symbols, where d is assumed to be even. Then, possible values of w can be written as

$$w = m\delta \quad \text{where} \quad m = -d, -(d-2), \ldots, 0, \ldots, d-2, d$$

when d is even and $m = -d, -(d-2), \ldots, -1,\ 1, \ldots, d-2, d$ when d is odd. Furthermore, the number of signal/parity vectors sharing a given signal value $m\delta$ is given by $C_d^{\frac{d+m}{2}}$ using the familiar combinatorial notation. Figure 3-4 illustrates the need for grouping signal/parity vectors on a subcodeword of size $d = 2$.

The following change of representation from the signal/parity format to the signal/type-count format allows the convenience of recording a single vector per signal value present regardless of the number of underlying information symbol patterns. The new representation assigns a *type* to every possible parity pattern, where the type of a pattern is simply defined as the decimal equivalent of the corresponding binary string. Consequently, there are 2^m possible parity types, ranging from 0 to $2^m - 1$. When grouping several signal/parity vectors associated with the same signal value w, it then becomes sufficient to record the number of different parity types present to preserve all information prior to grouping. The vector $\mathbf{t} = (t_0, t_1, \ldots, t_{2^m-1}) \in \mathbb{N}^{2^m-1}$ associated with some possible signal value w is referred to as the *parity type vector.* The corresponding vector elements t_j, for $j = 0, \ldots, 2^m - 1$, indicate the number of the information symbol patterns, out of the 2^d possible patterns, yielding the signal

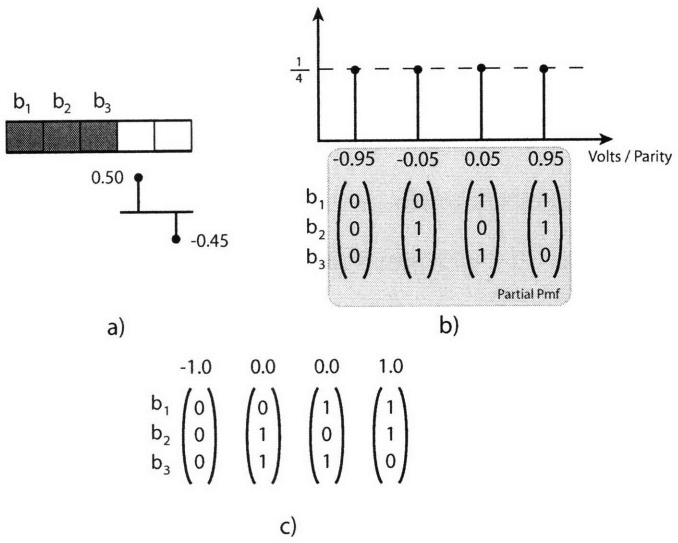

Figure 3-4: Example: The motivation for grouping subcodeword information vectors with identical signal values– a) The subcodeword and the corresponding portion of the channel response. b) The combinatorial expansion and the signal/parity vectors. c) The quantized signal/parity vectors where the quantization step is $\Delta = 0.1$. As expected, there are $C_2^1 = 2$ vectors associated with $w = 0$.

contribution w and the parity contribution of type j. Note that

$$\sum_{w \in \Omega} \sum_{j=0}^{2^m - 1} t_j^{(w)} = 2^d$$

where Ω represents the set of the possible values of w for a given subcodeword and the elements of the parity type vector $\mathbf{t}$ are endowed with superscripts to distinguish the vectors corresponding to distinct signal values. Finally, note that the parity type vector together with w forms a *coordinate*. Figure 3-5 illustrates the mechanism of switching to the new representation and grouping the two resulting coordinates.

Combining

Once the partial information is computed for each subcodeword, it is necessary to combine the results to yield the probability mass function corresponding to the full codeword, f_{W_i}. The combining is a two-step process. First, two subcodewords, each associated with d information symbols, are combined to yield the equivalent partial information for the resulting $2d$ symbols. Second, once all the subcodewords are

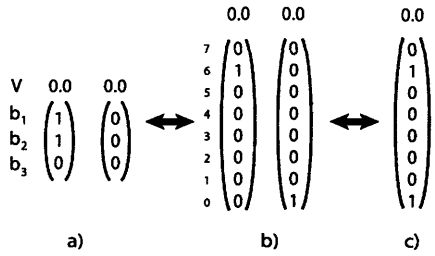

Figure 3-5: Example: Grouping several signal/parity vectors with the same signal value – a) Possible parity types for $k = 2$. b) Transforming two signal/parity vectors to reflect their parity patterns. c) Grouping the vectors into one coordinate.

combined, it suffices to transform the resulting parities into signal values and convert the information given in the coordinate form into the typical signal/probability representation. The two steps are described subsequently.

Since the subcodewords are formed by non-overlapping blocks of information symbols, two subcodewords are effectively independent. It follows that any combination of two coordinates associated with two different subcodewords is allowed. In other words, assuming the partial information for two subcodewords is composed of M and N coordinates respectively, the equivalent contribution of $2d$ information symbols results in $M \times N$ coordinates. Note that these need not be distinct, so the combining step is again followed by grouping coordinates with identical signal values.

To better describe the basic combining principle, it is convenient to briefly revert to the original signal/parity vector representation, prior to any grouping. A given pattern of d independent bits from subcodeword r produces a signal contribution w_r and a parity state $(b_{r1}, b_{r2}, ..., b_{rm})^T$. For two signal/parity vectors belonging to two distinct subcodwords, the task at hand is to combine the vectors so that the result is identical[7] to a signal/parity vector of the equivalent $(2d)$-symbol-long subcodeword. Represent the two coordinates as $(b_{r,1}, b_{r,2}, \ldots, b_{r,m})^T$ associated with w_r and $(b_{s,1}, b_{s,2}, \ldots, b_{s,m})^T$ associated with w_s, corresponding to subcodewords r and s respectively. Then, the signal/parity vector resulting from the combination is simply

[7]Ignoring any quantization issues, to be discussed shortly.

given as

$$(b_{r,1} \oplus b_{s,1}, b_{r,2} \oplus b_{s,2}, \ldots, b_{r,m} \oplus b_{s,m})^T \text{ associated with } w_r + w_s \tag{3.9}$$

where $\oplus$ denotes a *bit-wise xor* operation. In other words, to obtain the equivalent contribution of the $2d$ information symbols to both the received signal and the parity, it sufices to *add* individual d-symbol contributions in both the parities and in the signal values, making sure to define the notion addition with respect to the underlying field. An example of the combining principle on the signal/parity vectors is shown in Figure 3-6 for a simple single parity check code.

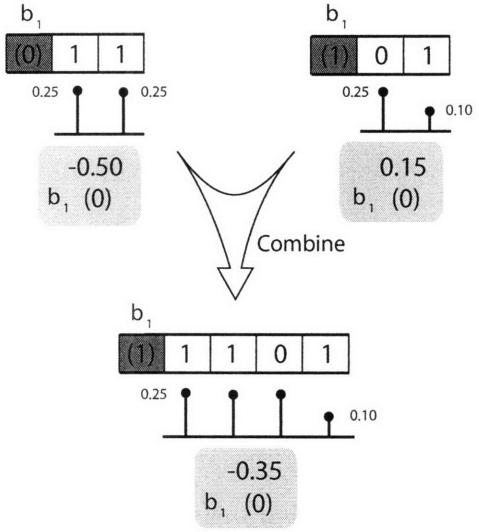

Figure 3-6: Example: Combining two signal/parity vectors corresponding to two different subcodewords to yield the equivalent signal/parity contribution – a) The original subcodewords with $d = 2$ implementing a single parity check code (SPC). b) The equivalent subcodeword with $2d$ information symbols implementing the same single-parity check code. In all three cases, the signal/parity vector has only two elements, associated with the signal contribution and the contribution to the single parity bit, respectively. The corresponding values are highlighted for clarity.

Extending the above result to the grouped coordinate representation relies on the fact that there are 2^m distinct combinations that can produce a given parity pattern. For instance, a pattern $(b_1, b_2, b_3) = (0, 0, 0)$, i.e. the parity of *type 0*, can be obtained by combining $(0, 0, 0)$ with $(0, 0, 0)$, as well as $(1, 1, 1)$ with $(1, 1, 1)$, or in general by combining any two identical parity patterns. Similarly, the parity pattern of *type 1* is

produced when combining any two patterns $(b_{r,1}, b_{r,2}, b_{r,3})$ and $(b_{s,1}, b_{s,2}, b_{s,3})$ such that $(b_{r,1} \oplus b_{s,1}, b_{r,2} \oplus b_{s,2}, \ldots, b_{r,m} \oplus b_{s,m}) = (0, 0, 1)$. The general result can be formalized as follows. Define the distance $d : \{-1, 1\}^m \times \{-1, 1\}^m \mapsto \mathbb{N}$ between two patterns as

$$d((b_{r,1}, b_{r,2}, b_{r,3}), (b_{s,1}, b_{s,2}, b_{s,3})) = \mathrm{dec}(b_{r,1} \oplus b_{s,1}, b_{r,2} \oplus b_{s,2}, \ldots, b_{r,m} \oplus b_{s,m})$$

where the dec operator converts an unsigned binary string into its equivalent decimal integer representation. Then, a parity pattern of *type* j, where $0 \leq j \leq 2^m - 1$, is produced by combining any two patterns at a distance j from each other.

Since following the grouping of the signal/parity vectors, each resulting coordinate may be associated with more than one parity pattern, when combining two coordinates from distinct subcodewords it is necessary to consider all the parity combinations which yield a given *type*. From Eqn 3.9, it is apparent that for any *type* j and any pattern $\mathbf{b}$, there is a unique counterpart $\mathbf{b}'$ such that $\mathbf{b} \oplus \mathbf{b}' = \mathrm{bin}(j)$, where bin operator translates an integer taking values in the range $\{0, \ldots, 2^m - 1\}$ into an unsigned binary string of length m. Given some parity type j, the ordered set of such counterparts, or, more precisely, of their decimal equivalents, is termed *the permutation vector for type* j, denoted $\mathbf{I}_j$. For reasons to become apparent shortly, the i^{th} element of the permutation vector $\mathbf{I}_j$ is given by

$$\mathbf{I}_j(i) = \mathrm{dec}(\mathrm{bin}(i) \oplus \mathrm{bin}(j)) \quad \text{for } i = 0, 1, \ldots, 2^m - 1\,,$$

where $\oplus$ is the bit-wise *xor* operator. Permutation vectors for $m = 3$ and the resulting 2^m possible parity types are shown in Table 3.1. This choice of terminology is indicative of each permutation vector's purpose. More precisely, consider the combination of two signal/parity coordinates

$$\mathbf{t}_r = (t_{r,0}, t_{r,1}, \ldots, t_{r,2^m-1})^{\mathrm{T}} \text{ associated with } w_r$$

with

$$\mathbf{t}_s = (t_{s,0}, t_{s,1}, \ldots, t_{s,2^m-1})^{\mathrm{T}} \text{ associated with } w_s$$

corresponding to codewords r and s respectively. Then, the number of resulting patterns of *type 0*, denoted by $t_{rs,0}$, is given as

$$t_{rs,0} = \mathbf{t}_r^{\mathrm{T}} \mathbf{t}_s.$$

In other words, the result is an inner product of two parity vectors. Extending this principle to other parity types is achieved by preceding the inner product by a vector permutation. Specifically, the number of resulting patterns of *type j* is given by:

$$t_{rs,j} = \mathbf{t}_r^{\mathrm{T}} \times \mathbf{P_j} \times \mathbf{t_r} \tag{3.10}$$

where the $\times$ operator indicates ordinary matrix multiplication. In the above equation, $\mathbf{P_j}$ denotes a $(2^m - 1) \times (2^m - 1)$ permutation matrix with ones at positions

$$(i, \mathbf{I}_j(i)) \quad \text{for} \quad i = 0, \ldots, 2^m - 1,$$

and zeros elsewhere. Note that, as previously, $\mathbf{I}_j(i)$ denotes the i^{th} element of the permutation vector $\mathbf{I}_j$. It follows that the combining operation can be efficiently implemented as a series of 2^m permutations and dot products. This is illustrated in Figure 3-7 where the detailed computation is shown for the parity pattern of *type 2*.

$\mathbf{I}_0$	$\mathbf{I}_1$	$\mathbf{I}_2$	$\mathbf{I}_3$	$\mathbf{I}_4$	$\mathbf{I}_5$	$\mathbf{I}_6$	$\mathbf{I}_7$
7	6	5	4	3	2	1	0
6	7	4	5	2	3	0	1
5	4	7	6	1	0	3	2
4	5	6	7	0	1	2	3
3	2	1	0	7	6	5	4
2	3	0	1	6	7	4	5
1	0	3	2	5	4	7	6
0	1	2	3	4	5	6	7

Table 3.1: Permutation vectors for $m = 3$. Each permutation vector $\mathbf{I}_j$ is the *type j* counterpart to $\mathbf{I}_0$.

Once all the subcodewords have been combined, it remains to transform the resulting set of coordinates into the typical signal/probability representation. Since, for any given coordinate, the signal value w represents the contribution to W_i from

the codeword's information bits, it suffices to transform the remaining parity bits into their equivalent signal contribution and add the latter to w. The pattern count associated with each parity type, divided by the total number of possible symbol patterns, 2^k, is then used to compute the corresponding probabilities. This is further illustrated in Figure 3-8.

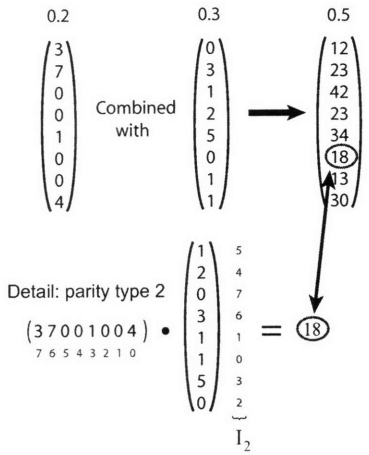

Figure 3-7: Example: Combining two coordinates from two different subcodewords with $m = 3$. The principle of the record permutation is shown in more detail for the parity pattern of *type 2* where $\mathbf{I}_2$ is used to compute the resulting probability counts.

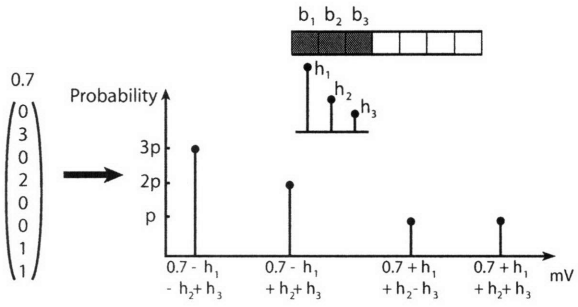

Figure 3-8: Example: Translating the signal/parity-type coordinates into a standard probability distribution. The conversion is illustrated for one coordinate. The parity patterns present are converted to PAM2 (here, '0' $\rightarrow +1$) and mapped to the corresponding channel coefficients as given in (2). The probability counts are extracted from the field associated with each type and normalized by $p^{-1} = 2^k$, corresponding to the the total number of possible bit patterns associated with one codeword.

3.4.2 Generalizations and Remarks

This section addresses the complexity and quantization issues in more detail. It also extends the scope of the previously-described algorithm to dealing with different codeword symbols, accounting for co-channel interference and estimating the performance of codes over higher fields

Extension to Multiple Codewords and Different Symbol Locations

The previous section describes the method of dealing with a simplified scenario illustrated in Figure 3-1 and further reproduced in Figure 3-9a). In the simplified scenario, the channel response spans one full codeword, corresponding to symbols $X_{i-1}, \ldots, X_{i-L+1}$, while the most recently transmitted symbol, denoted by X_i, corresponds to the first symbol of a new codeword. From Equation 3.1, the received signal Y_i is given by

$$Y_i = h_0 X_i + h_1 X_{i-1} + \ldots + h_{L-1} X_{i-L+1}.$$

The described algorithm focuses on computing probability distribution for the quantity $W_i = h_1 X_{i-1} + \ldots + h_{L-1} X_{i-L+1}$. Subsequently accounting for the value of X_i is straightforward since the symbols belonging to two distinct codewords are statistically independent.

A simple extension for dealing with multiple full codewords, illustrated in Figure 3-9b) relies on the statistical independence between codewords. More precisely, consider the received signal at time i and let $W_i^{(1)}, W_i^{(2)}, \ldots, W_i^{(K)}$ be the contribution to Y_i from the K most recently transmitted codewords[8] so that

$$Y_i = h_0 X_i + W_i^{(1)} + W_i^{(2)} + \ldots + W_i^{(K)}.$$

Since different codewords are statistically independent, so are their contributions to the received signal. It therefore follows that the probability distribution $f_{Y_i|X_i=1}$ is given as the convolution of the probability distributions corresponding to each

[8]Note that, for some (n, k) linear block code, if the channel length is not an integer multiple of the codeword length n, the tail of the channel response can be zero-padded.

$W_i^{(j)}, j = 1, \ldots, K$, where the abscissa of the resulting distribution is shifted by h_0 to account for the conditioning. In turn, the probability distributions for each $W_i^{(j)}, j = 1, \ldots, K$ are obtained through the algorithm described in the previous section.

Allowing $W_i^{(1)}$ to be based on an incomplete codeword provides means of computing marginal probability distributions for any symbol location in a codeword. The probability distribution governing $W_i^{(1)}$ is also obtained using the proposed algorithm by zero-padding the corresponding portion of the channel response to form a full codeword. However, in order to condition on the value of the most-recently transmitted symbol, that is, conditioning on $X_i = x$ for some $x \in \{-1, 1\}$, one modification is necessary. Specifically, when computing the partial information corresponding to the most recent subcodeword, both the sequential and the batch approach need to consider only the symbol patterns where $X_i = x$. Finally, since the codeword independence still holds, the final result is once again given as a convolution. The overall process is illustrated in Figure 3-9c).

Computational Complexity

Note that, prior to discussion possible extensions to higher-order alphabets, it is useful to consider the computational complexity of the algorithm. For some given codeword length n, increasing the length of the channel response, L, increases the number of previously transmitted codewords spanned by the channel memory. This translates into a linear increase in the number of convolutions of PMFs corresponding to individual codeword to the total received signal. Similarly, for a codeword with a given number of parity bits m, increasing the number of independent bits k linearly increases the number of subcodewords of of size d that need to be considered. The increase in the number of subcodewords has no effect on the grouping or combining algorithms, and the overall complexity increase is therefore linear as well. However, the increase in complexity due to increasing the number of parity bits, m, is exponential. This is due to the parity tracking mechanism necessary in the grouping step of the algorithm, or, more precisely, to the necessity of translating an m-symbol parity pattern to 2^m

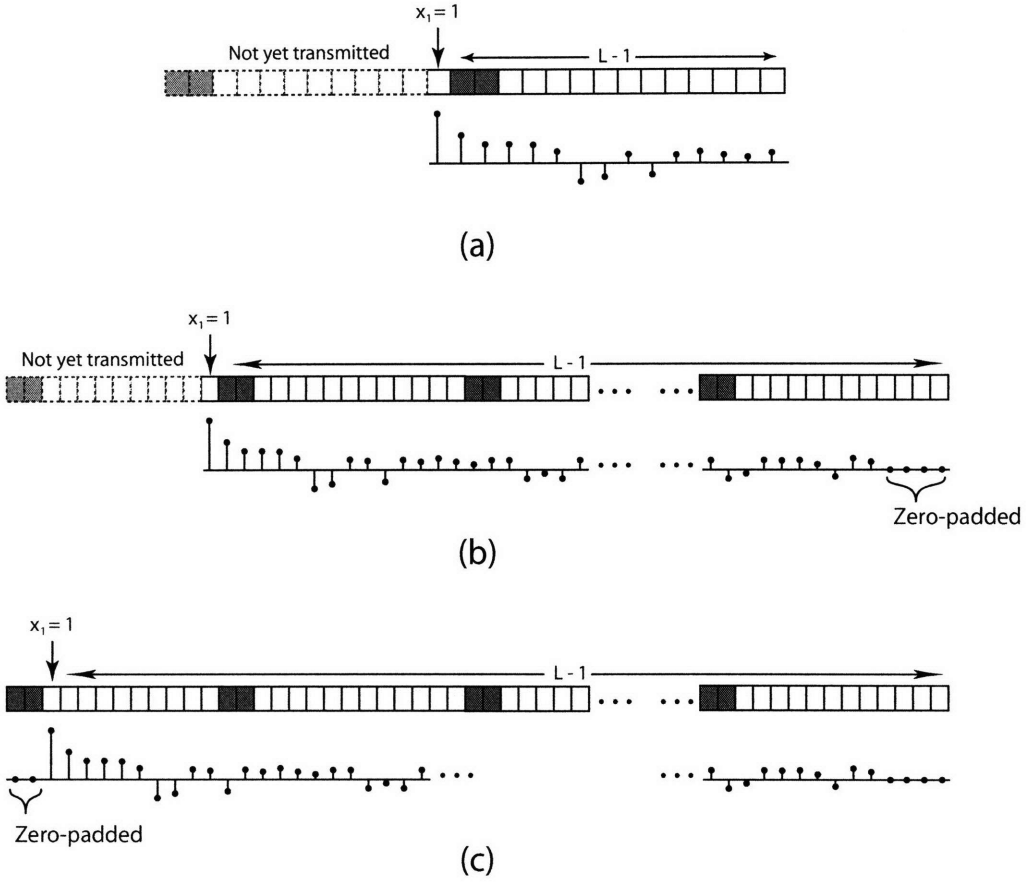

Figure 3-9: Extension to multiple codewords and symbol locations – a) The simplified scenario assumed in the previous sections. b) The extension to multiple codewords. c) The extension to different symbol locations withing a codeword. Note that the channel response is depicted as having no pre-cursor coefficients. However, the latter can be accounted for in an analogous manner.

parity pattern types. Nevertheless, for high-rate codes, the number of parity bits is typically much less than the number of independent bits and the overall reduction in complexity is still significant. Section 3.5 illustrates the runtime performances for different codes based on a MATLAB implementation of the technique.

Extension to Higher-order Alphabets

Although the previous developments concerned codes over binary alphabets, the new algorithm also extends to higher-order alphabets. Assuming an alphabet of size q the resulting parity-type vectors are of size $q^m - 1$. An analogous rule can be developed to determine the corresponding permutation vectors. However, the principal difficulty of accounting for higher-order alphabets is the corresponding complexity increase. Assuming $q = 2^l$, the complexity now increases as $q = 2^{ml}$ where m is the number of parity bits. Thus, while given sufficient memory resources, results can be obtained with $q = 4$ and $m = 10$ and any codeword length, higher-order alphabets or additional parity bits quickly become computationally prohibitive.

Quantization

In order to reduce the computational requirements, and therefore the execution time of the algorithm, it is beneficial for the signal contributions to be quantized. This does not reduce the size of the parity-type vector, but it reduces the number of signal/parity coordinates that need to be recorded.

However, instead of quantizing the channel coefficients directly, quantizing the signal contributions at the subcodeword level, prior to grouping, significantly reduces the resulting quantization error. More precisely, given a subcodeword of d information symbols spanned by some set of channel coefficients, quantizing each channel coefficient to some precision Δ incurs a quantization error distributed over the interval $[-d\frac{\Delta}{2}, d\frac{\Delta}{2}]$. For the same Δ, quantizing instead the signal contribution due to d, the corresponding interval shrinks to $[-\frac{\Delta}{2}, \frac{\Delta}{2}]$.

The total quantization error for the full probability distribution depends on the quantizing Δ and the ratio of the channel length L to the subcodeword size d. Quan-

tizing only at the level of signal contributions for each subcodeword, as previously discussed, then causes a total quantization error of at most $\lceil L/d \rceil \frac{\Delta}{2}$. It is therefore advisable to assign a large, but practical, value to d. As an illustration, consider a system with the channel length $L = 200$ and some (100,90) systematic linear block code. It follows that the channel response spans at most two codewords. In addition, let $\Delta = 10^{-4}$ (0.1 mV) and $d = 10$. Each codeword is therefore subdivided into 9 subcodewords and the resulting maximum quantization error for the total received signal becomes 9×10^{-4}. Assuming the distribution of the quantization error to be symmetric around 0 and individual quantizing errors for each signal to be independent, the maximum quantization error, based on the probability that quantization errors from all 18 subcodewords interfere positively or negatively, occurs with probability 2^{-17}. More realistic bounds on the quantization error can be obtained by using any of the well-known bounds for sums of independent identically distributed random variables with finite variance.

3.5 Practical Examples

The previous section introduced an efficient algorithm that allows to accurately quantify the effect of code constraints, in the form of a systematic linear block code, on the decision distance, regardless of the operating regime. This chapter provides further illustrations regarding possible algorithm implementations and the information the corresponding tool provides in a typical high-speed link. Since the practical examples of the previous chapter provide a thorough illustration of the behavior of codes in the (quasi-)worst-case-dominant regime, this section focuses on the large-set-dominant scenario.

All the results are obtained with the equalized channel responses, B3 and B32 [17], shown in Figure 3-10. Both channels display a long pulse response, on the order of 300 coefficients. However, unlike its counterpart, Channel B32 contains dominant interference coefficients, caused by insufficiently equalized dispersion.

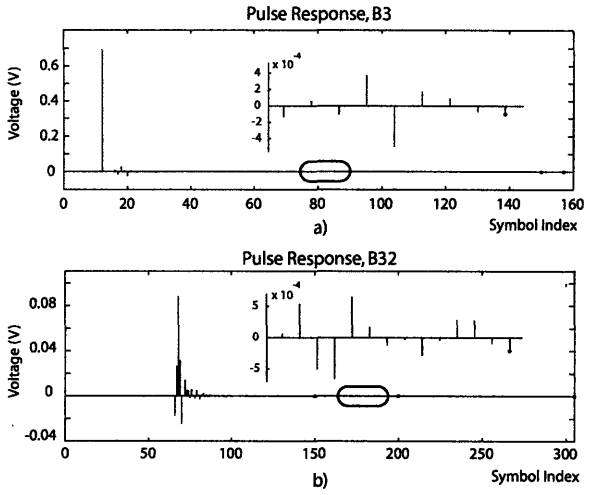

Figure 3-10: Two High-speed Link Channels – The equalized pulse responses shown are based on the Peters ATCA measurement data for a 3" [top, B3] and 32 " [bottom, B32] backplane with bottom-layer routing [17]. The zero-forcing equalizer has 5 coefficients. The links operate at 5 Gb/s and 10 Gb/s respectively.

3.5.1 Runtime Statistics

All the distributions computed in this section are obtained with $d = 10$ bits and $\Delta = 10^{-4}$ Volts. The runtime statistics were obtained on a Pentium 1.60 GHz processor with 504 MB of RAM. Table II a) presents the runtimes for Hamming codes of different sizes. For codewords whose length exceeds the length of the channel, the channel response is zero-padded. As this increases the effective channel length the runtime results present a pessimistic view of the efficiency of the simulator for particularly long codewords. To illustrate the increase in the runtime incurred by an increase in the number of parity bits alone, Table 3.2b) provides the runtime statistics for an arbitrary $(290 + m, 290)$ code.

Code (n,k)	Runtime (s)	Code (n,k)	Runtime (s)
(31,26)	5.15	(295,290)	11.0
(63,57)	9.86	(296,290)	12.8
(127,120)	15.5	(297,290)	18.4
(255,247)	38.0	(298,290)	36.0
(511,502)	178	(299,290)	104
(1023,1013)	1.04×10^3	(300,290)	453

Table 3.2: Runtime statistics – a) Hamming codes. b) Instances of random codes of fixed codeword length.

Note that, for MATLAB implementations, the above runtimes can be further decreased by taking advantage of sparse matrices. The corresponding benefit has less to do with the sparsity of the data, but provides instead a more efficient implementation of the combining process through facilitated indexing.

3.5.2 Link Performance

The following considers the effect of code constraints on marginal symbol error probabilities of the two high-speed link channels of interest. The corresponding joint error statistics are also discussed.

Effect of Constraints on Symbol Error Probabilities

To illustrate the effect of restricting the transmit alphabet through some systematic linear block code on the decision distance, the results of this section are based on a hypothetical set of codes implemented in a high-speed link. Choosing the set of all possible (10,8) systematic linear block codes is motivated by the runtime-limiting/DC-balancing 8b/10b codes [18] traditionally implemented in some high-speed links. While the 8b/10b codes are not systematic and therefore cannot be simulated through the proposed framework, it is interesting to consider the effects of possible code constraints, in the form of a systematic linear block codes of identical overhead, on the probability distribution of the random variable Z_i. The resulting probability distributions are shown in Figure 3-11. The effects of adding a realistic amount of system noise to these distributions are shown in Figure 3-12.

Prior to discussing the above figures, several remarks are in order. First, note that different operating points are obtained by shifting the decision threshold to artificially increase the decision distance. Although Section 2.5.1 of the previous chapter cautions against this practice in performance estimation, its use in the present context is justified. More precisely, while this practice can lead to misleading results when biasing the operating conditions in order to improve the accuracy of the performance estimate for a given system, its present use allows to quickly illustrate the performance

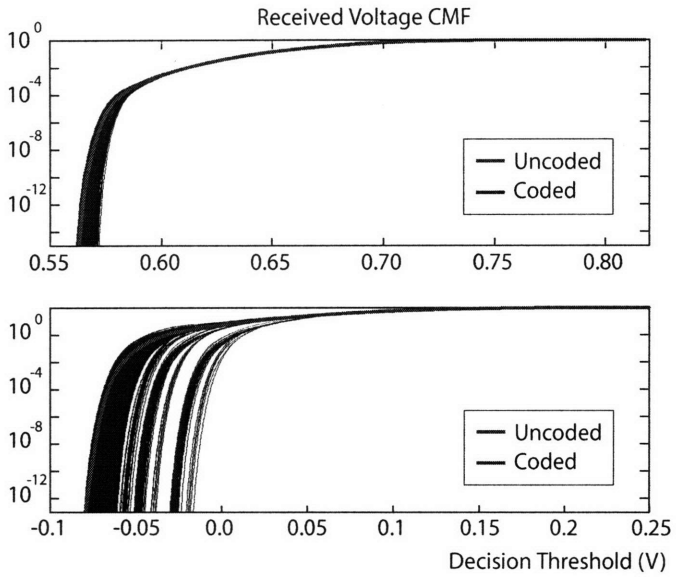

Figure 3-11: Received signal distributions for the set of all (10,8) linear block codes. Shown are the cumulative mass functions (CMF) for two different channels: a) B3 b) B32. The plots also show the signal CMF computed under the assumption that the data is uncoded. For consistency, both the coded and the uncoded CMF are quantized $d = 4$ independent bits at a time. The quantization step is 0.5mV. The probability distributions shown correspond to possible signals associated with the first information symbol in a codeword.

of *different* systems sharing the same channel signature. Also note that, for each channel, the symbol error probabilities are computed for two codes: *code 1* and *code 2* for channel B3 as well as *code 3* and *code 4* for channel B32. The codes are chosen arbitrarily from the set of all (10,8) codes in order to illustrate the situations where a code can make a moderate to large difference in the marginal symbol error probabilities.

The corresponding results show that different (10,8) systematic binary linear block codes can drastically change the interference profile for a given symbol, and with it, the symbol error probability. In addition, this behavior can differ greatly from the behavior of an uncoded system. The corresponding symbol error probabilities can differ by an order of 10^3 for the B3 channel operating at an uncoded BER of 10^{-10} and $>> 10^{21}$ for the B32 channel under the same conditions. The resulting discrepancy in the probability distributions between the coded and the uncoded system represents the error in computing the coded probability distributions assuming an unconstrained stream of symbols. Alternatively, this discrepancy also quantifies a possible benefit

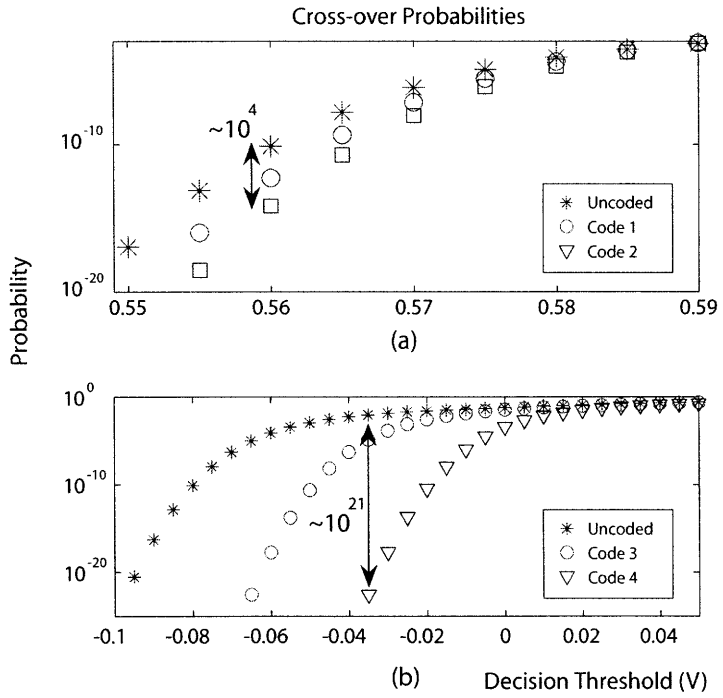

Figure 3-12: Bit cross-over probabilities for the uncoded case and two different (10,8) linear block codes on a B3 channel [top] and B32 channel [bottom]. Code 2 was chosen to yield the maximum deviation from the uncoded CMF, while Code 1 was chosen roughly in between the two extremes. The results are computed under the common assumption that the link noise is additive, white and Gaussian with $\sigma \approx 3$mV,

of prefering certain codes based on their pattern-eliminating properties.

The a priori and a posteriori distributions for both channels and different threshold values are shown in Figures 3-13 and 3-14, respectively. Note that, while the B3 channel is neither worst-case nor quasi-worst-case dominant, the B32 channel is quasi-worst-case dominant throughout the relevant threshold range. More precisely, placing the decision threshold at -0.01 results in the value[9] $1 - \tilde{f} = 0$. On the other hand, placing the decision threshold at -0.04 yields $1 - \tilde{f} = 2.7 \times 10^{-6}$. In the latter case, the corresponding uncoded error probability is $p_{err} = 4.3 \times 10^{-3}$. The discrepancy between $p_{err} \times 10^{-6}$ and the improvemement of 10^{-21} depicted in Figure 3-12 is due to the fact that *code 2* prohibits several harmful patterns, in addition to the worst-case pattern, from occuring on the first symbol in the codeword. Since the dispersion coefficients

[9]Since the noise has infinite support $1 - \tilde{f} > 0$. The lack of significant digits is due to finite numerical precision.

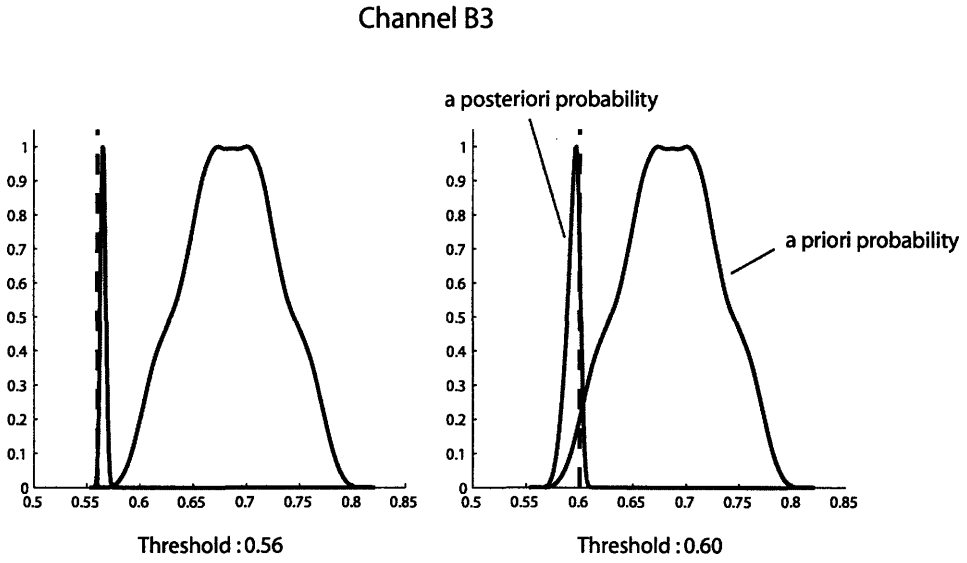

Figure 3-13: A priori and a posteriori probabilities for the random variable Z_i over the B3 channel. The probabilities correspond to an uncoded system with $\sigma = 3$ mV. The system does not operate in a worst-case-dominant regime as the noise variance is large compared to the spacing between any two coefficients. On the other hand, since there are no dominant interference coefficients, the channel contains no primary part and the system cannot operate in the quasi-worst-case-dominant regime either.

are large compared to the noise standard deviation, also prohibiting the next-to-worst-case events has a dramatic effect on the resulting symbol error probability.

However, note that despite the apparent error reductions, the performance of the above codes is inferior to that of a pattern-eliminating code. Specifically, pattern-eliminating codes provide an error reduction for *all symbols in a codeword*, while the present results depict the maximum benefit of a (10,8) code focusing on the symbol error probability of the *first codeword symbol* alone. In fact, repeating the experiment focusing on the second symbol in the codeword yields very different results: *code 2* and *code 4* no longer provides a significant benefit over the uncoded case.

Joint Statistics

Given the accurately computed symbol error probabilities, it remains to determine the joint error statistics for different symbols in a codeword. For systems operating in the quasi-worst-case-dominant scenario, the errors occuring on distinct symbols are in general *not independent*, as demonstrated in the previous chapter. Specifically,

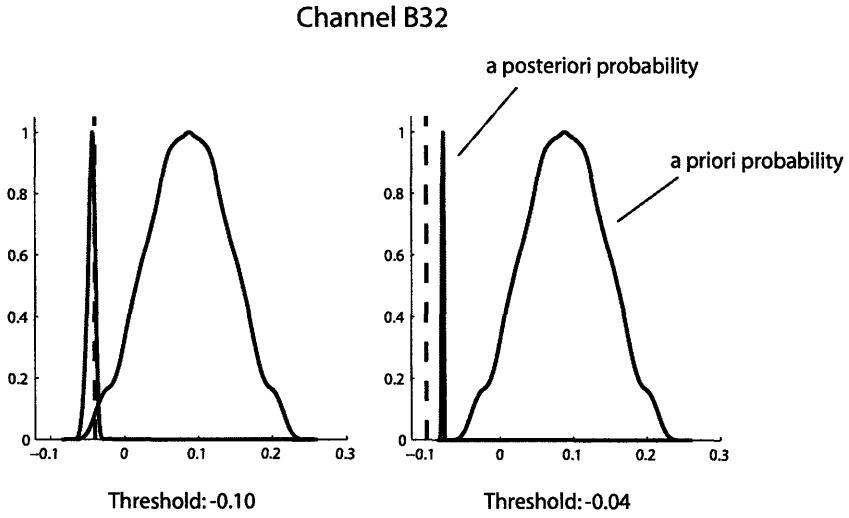

Figure 3-14: A priori and a posteriori probabilities for the random variable Z_i over the B32 channel. The probabilities correspond to an uncoded system with $\sigma = 3$ mV. The system operates in the quasi-worst-case-dominant scenario, where the primary channel spans five symbols, including the pre-cursor interferers. The corresponding worst-case parameters are $1 - \tilde{f} = 0$ [left] and $1 - \tilde{f} = 2.7 \times 10^{-6}$ [right].

Section 2.5 of Chapter 2 considers the performance of the single parity check code for a system operating in the quasi-worst-case-dominant scenario and illustrates the discrepancy between the measured performance and that predicted based on the error independence assumption. Due to the similarity of the B32 channel to the dispersive channel considered in Chapter 2, the analogous results for the B32 channel are not displayed.

The variance of the ISI for the B3 channel approximately equals 1.1×10^{-3}, which is significantly greater than the noise variance $\sigma^2 = 9 \times 10^{-9}$. Therefore, the system does not operate in the large-noise scenario, as further evidenced by the corresponding a posteriori probability plots of Figure 3-13. Nevertheless, since the system is neither worst-case nor quasi-worst-case dominant, the results of Section 2.3.2 of Chapter 2 suggest that the error events can be still considered as independent up to a certain accuracy. In fact, the corresponding results suggest that independence assumption is relatively accurate for long, well-equalized channels. However, the underlying argument assumes the transmissions to be uncoded, while accurate joint error statistics are particularly important in performance estimation of standard error control codes.

Moreover, the argument provides little indication of the allowed range of system parameters so that the independence assumption holds with some given confidence.

In order to experimentally quantify the validity of the assumption for the B3 channel in the above operating conditions, Tables 3.3 and 3.4 show the predicted and measured word error rates (WER) for the (31,26) Hamming code and an extended Golay code, respectively. The previously described algorithm is used to analytically compute the cross-over probabilities, p_{err}, recorded in the "predicted" field. The cross-over probabilities are computed using $\Delta = 10^{-4}$V and $d = 10$, and are averaged over all the symbol locations within the codeword. Since the Hamming code successfully corrects all single-bit errors, the corresponding word error rate is predicted according to the familiar expression

$$WER|_{Ham} = 1 - (1-p)^{31} - 31p(1-p)^{30}.$$

Similarly, the WER for the Golay code is given by

$$WER|_{Gol} = 1 - (1-p)^{24} - 24p(1-p)^{23} - 276p^2(1-p)^{22} - 2024p^3(1-p)^{21}$$

where the decoding error occurs if more than 3 bit errors are present in the codeword. The Monte Carlo simulator operates with 10^9 codewords.

Decision	Predicted		Measured	
Threshold (V)	p_{err}	WER	p_{err}	WER
0.56	1.45e-8	9.70e-14	1.32e-8	0
0.57	1.81e-5	1.53e-7	1.80e-5	1.30e-7
0.58	5.10e-4	1.20e-4	5.09e-4	2.68e-4
0.59	3.16e-3	4.38e-3	3.16e-3	7.56e-3
0.60	1.12e-2	4.67e-2	1.11e-2	5.36e-2

Table 3.3: Predicted and measured cross-over probabilities (p_{err}) and word error rates (WER) for a (31,26) Hamming code

Decision	Predicted		Measured	
Threshold (V)	p_{err}	WER	p_{err}	WER
0.56	1.02e-8	1.20e-15	4.17e-8	0
0.57	2.33e-5	2.65e-15	2.21e-5	0
0.58	4.77e-4	5.47-10	5.20e-4	0
0.59	3.22e-3	1.08-6	3.17e-3	4.00e-6
0.60	1.12e-2	1.40e-4	1.12e-2	6.31e-4
0.61	2.79e-2	4.11e-3	2.76e-2	1.00e-2

Table 3.4: Predicted and measured cross-over probabilities (p_{err}) and word error rates (WER) for an extended Golay code, (24,12)

The above results suggest that, at word error rates that can be accurately captured by Monte Carlo simulation, the error independence assumption yields adequate[10] WER approximations for both codes. Note that this occurs despite the fact that, for the extended Golay code, exactly half of the symbols in each 24-symbol codeword are constrained. Furthermore, the independence assumption is relatively accurate despite the fact that the a posteriori probability is concentrated at signal values relatively far from the signal mean, as evidenced by the a posteriori probability distributions of Figure 3-13. The accuracy is a consequence of the well-equalized link channel reponse, where the length of the response is sufficiently large to provide a large multiplicity for the corresponding signal values, while the magnitude of the interference coefficients is sufficiently small to reduce the effect of the discrepancy between the present conditions and the idealized case. As part of the future work, developing general expressions to specify, for a given level of accuracy, the magnitude of the interference coefficients, the channel length, and the amount of interference and/or redundancy the independence assumption can withold would provide the definitive answer to the question of performance estimation for coded high-speed links.

[10]The discrepancies between the predicted and the observed WER values are attributable to both the approximation error and the variance of the Monte Carlo estimate, where the latter is particularly pronounced at low error rates. For the cross-over probabilities, such discrepancies can result from the estimator variance and the quantization Δ.

3.6 Summary

The previous lack of systematic framework for code-space explorations motivated the development of performance estimation methods for coded systems with inter-symbol interference, described in this chapter. Given a measurement of the channel's impulse response, the noise distribution and some error-control code, the problem is that of accurately estimating the error rate of the system. The method of estimating the system's performance is tied to the underlying operating regime, and a simple computational method allows the latter to be determined given some detection scheme, equivalent channel pulse response and noise variance. For channels operating in the (quasi-)worst-case-dominant scenario, large-noise scenario, or in the limit of the the limit of large-set-dominant scenario, efficient methods of computing the joint error statistics are identified. In particular, for the (quasi-)worst-case-dominant scenario, it suffices to consider a small set of error-causing patterns, while in the large-set-dominant scenario, the errors are approximately independent. Note, however, that further work is required to quantify the accuracy of the independence assumption for a wider range of practical situations.

For any ISI-and-AWGN-limited system, including those that cannot be identified with one of the previous three scenarios with sufficient accuracy, an efficient algorithm is introduced to accurately compute marginal probability distributions for the received signal. Note that the same framework provides an easy extension to co-channel interference, which commonly occurs in high-speed links as near-end and far-end cross-talk (NEXT/FEXT [6]). In the future work, the timing jitter can also be addressed by first conditioning the signal distributions on either the likely symbol patterns or the corresponding ISI values, and augmenting those by the corresponding jitter distributions. In that sense, the timing and the interference problems are separable.

Conclusion

Modeling a high-speed link as an ISI-limited system with additive white Gaussian noise allows for an abstracted framework suitable for a more theoretical approach to studying the benefit of coding for high-speed links. Possible error mechanisms are categorized according to three regimes — the large-noise, the large-set-dominant and the worst-case-dominant — which are entirely specified by the system's noise level and the channel's pulse response. In the large-noise and large-set-dominant regimes, classical coding theory provides an exhaustive characterization of different error-control codes, whose hardware complexity has already been partially addressed in [4]. While the worst-case-dominant regime occurs rarely in a high-speed link, the quasi-worst-case-dominant regime is shown to occur on ATCA channels at realistic equalization levels and is also shown to be consistent with previous experimental observations. However, implementing standard error-control codes in the quasi-worst-case-dominant regime generally leads to a negative result, as in the case of [4], or an inconsistent result, as reported in [12]. This thesis further examines the behavior of standard error-control codes in this regime and shows that for uncorrelated channels, where the notion of correlation is redefined through nesting of the worst-case symbol patterns, a single parity check code is optimal, as long as the codeword length is inferior to the correlation length. Conversely, for correlated channels, including channels with insufficiently equalized dispersion, a standard error-control code requires a potentially large overhead due to an increased probability of observing a relatively large number of errors in a codeword, conditioned on the occurrence of a worst-case symbol pattern.

For systems operating in a worst-case-dominant or the quasi-worst-case-dominant regime, this thesis develops a more efficient approach to coding, based on systemati-

cally eliminating all occurrences of worst-case symbol patterns. The resulting codes, referred to as the pattern-eliminating codes, are systematic block codes that require virtually no decoding. In particular, the $(n, n-1)$ pattern-eliminating codes also allow for simple encoding and are effective for many channels of interest, including the dispersive channel, while the more complex $(n, n-2)$ codes are effective over all communication channels. The simulation results show that an $(n, n-1)$ pattern-eliminating code can virtually eliminate all errors[11] on realistic high-speed link channels. However, such benefits often vanish when the rate penalty of a code is taken into account. A method of overcoming the rate penalty of a pattern-eliminating code is to endow the code with additional timing benefits. Over most channels, an $(n, n-1)$ pattern-eliminating code can be extended by a simple rule to also perform run-length-limiting, with the maximum allowed runlength of $n-1$. In that sense, a $(6, 5)$ pattern-eliminating code provides the run-length-limiting capability of the 8b/10b code [18], commonly implemented in high-speed links, but incurs less overhead. Furthermore, the $(6, 5)$ pattern-eliminating code, coupled with a suitable equalization method, is also shown to provide an error reduction of over fifteen orders of magnitude over a realistic high-speed link channel. Thus, although the pattern-eliminating code does not provide the DC balancing property of the 8b/10b code, the significant reductions in the implementation complexity, overhead and error rate inherent to pattern-eliminating codes are, in principle, often sufficient to render the latter strongly preferable for run-length-limiting applications in high-speed links.

However, further work is required on extending the pattern-eliminating properties to deal with a wider range of operating conditions. In particular, one of the remaining problems consists of identifying or developing suitable equalization or channel conditioning techniques that optimize the performance of a pattern-eliminating code. Such equalization is, in principle, significantly more power-efficient compared to that employed in current high-speed links, as the equalizer no longer needs to ensure a low error probability. The corresponding scheme could potentially yield significant benefits for high-speed links by enabling the communication at higher data rates than

[11] That is, provide error-rate reductions of over fourty orders of magnitude.

those achieved previously, or by providing the same signalling speeds at a greater energy efficiency. In addition, future developments should address the ability of a pattern-eliminating code to prohibit patterns from a larger set, as well as provide immunity against channel nulls[12]. Lastly, there may be a benefit to further investigating the ties between the pattern-eliminating codes and the constraint codes for magnetic recording systems. Although the two types of codes are shown to have largely different structures, some of the resulting insights pertaining to constraint coding over binary channels may be of use for extending the timing benefits and allowing for simplified encoder implementations of $(n, n-c)$ pattern-eliminating codes, when $c > 1$.

Finally, this thesis also addresses the previous lack of systematic framework for performance estimation and code-space explorations. The regime classification provides a more accurate guideline for biasing the system parameters in simulation to capture error behaviors at low probabilities. In addition, additional computational approaches that do not rely on any parameter biasing are identified for each of the regimes. Furthermore, for operating conditions that cannot be classified as one of the three regimes with sufficient accuracy, an efficient numerical algorithm is also introduced for the computation of marginal probability densities in systems with ISI implementing a systematic linear block code[13]. The three simulation methods can be used in conjuction to provide a reliable performance picture of a high-speed link, or a general system with noise and interference, operating in any regime. These performance estimation methods extend easily to near-end and far-end cross-talk (NEXT/FEXT [6]), while extensions to timing jitter can likely be addressed through a conditioning framework, and are relegated to future work.

[12]In the sense of channel coefficients of zero or small magnitude in the principal part of the channel response.

[13]The performance of pattern-eliminating codes is given by a closed-form expression, accurate over the range of conditions of interest.

Appendix - Proof of Theorem 9

Theorem 9. *The $(n, n-2)$ pattern-eliminating code is effective for any channel of finite length L and any $n \leq L$.*

The following lemma is an extension of Theorem 7 for the $c = 2$ case. To reduce the notational burden, the result is stated somewhat informally, but the accompanying illustration of Figure 3-15 renders the statement precise. The notation s_1, s_2, s_3 introduced in the figure to denote the shifts between consecutive nestings is employed in the subsequent theorem as well. Note that the four patterns are ordered by the position of the first symbol, so that $s_1, s_2, s_3 > 0$.

Lemma 3. *The $(n, n-2)$ pattern-eliminating code with $n = L$ is ineffective if and only if at least four symbol patterns of length L can be nested so that*

1. *the starting symbols of each pattern are all within a distance of $n-2$ of each other, as depicted (Figure 3-15—a);*
2. *there exists a "strip" of two consecutive symbols so that the starting symbol of each pattern occurs strictly to the left of the strip (Figure 3-15—b);*
3. *for each of the four patterns, the corresponding symbols form a worst-case pattern everywhere outside that strip (Figure 3-15—c);*
4. *the signatures of the channel coefficients, where the channel response is referred to the starting symbol of each pattern, that span the symbols within the strip exhaust the elements of the set $\{(1, 1), (-1, -1), (-1, 1), (1, -1)\}$ (Figure 3-15—d).*

Proof. It is clear that if one cannot nest four or more symbol patterns in this manner, it is always possible to pick the two constraint symbols so that each pattern is distant from the worst case by at least one symbol. Conversely, if there exists such a nesting, then aligning the location of the "strip" with the location of the constraint symbols yields a case where any choice of the constraint symbols yields to an occurrence of the worst-case pattern. □

Theorem 12. *There exists no nesting of four symbol patterns satisfying the conditions of Lemma 3.*

Proof. We show that any nesting satisfying the first three conditions cannot satisfy the last.

Step 1 First assume $s_1 > 1$ and consider the three symbol patterns labeled A, B and D in Figure 3-16—a. For the pattern A, let m denote the distance between the first information symbol and the strip. Equate the symbols of patterns A and B to the left of the strip. Since the patterns are in the worst-case everywhere outside the strip, the ability to nest A and B imposes a specific structure on the channel coefficients. Specifically, it follows that the first m channel coefficients are either periodic with period s_1 or periodic with period $2s_1$, but $c_j = -c_{j+s_1}$ for any $1 \leq i \leq m - s_1$. Note that the latter case is illlustrated in Figure 3-16—a).

Since $s_1 > 1$, we have that the channel coefficients spanning the strip in pattern A, namely c_{m+1} and c_{m+2}, span symbols to the right of the strip in pattern B (Figure 3-16—b). Equating the symbols of patterns B and D to the right of the strip, it follows that c_{m+1} and c_{m+2} take values from the first m channel coefficients. Furthermore, these coefficients are tied to c_{l+1} and c_{l+2}, where $l = m - s_1 - s_2 - s_3$ and c_{l+1} and c_{l+2} are the channel coefficients spanning the strip in pattern D. Specifically, the periodicity implies that c_{m+1} and c_{m+2} either equal c_{l+1} and c_{l+2} or equal the complement of c_{l+1} and c_{l+2}.

Similarly, considering the symbol patterns B, C and D and assuming $s_2 > 1$, it follows that the coefficients spanning the pattern B are equal or opposite to those spanning the pattern D. By the previous result, it follows that the channel coefficients spanning the constraint symbols of patterns A, B, and D can only exhaust two of the possibilities out of the set of four possible combinations, that is, $\{(\ 1,\ 1), (-1, -1), (-1,\ 1), (\ 1, -1)\}$. Thus, when $s_1 > 1$ and $s_2 > 1$, the last condition of Lemma 3 cannot hold.

Step 2 It remains to consider the case when either $s_1 = 1$ or $s_2 = 1$ or both. Suppose $s_1 = 1$. By an analogous argument to that of Lemma 1 (Chapter 2 Section 2.3), it follows that the first m symbols of pattern A are either all of the same sign or

alternating. Since it is always the case that $s_1 + s_2 \geq 2$, it follows that the channel coefficients spanning the constraint symbols of pattern C and D take values from the first m channel coefficients, and therefore also either of the same sign or alternating. Furthermore, also since $s_1 + s_2 \geq 2$, applying an analogous argument to that of Step 1 to patterns A, C and D shows that the channel coefficients spanning the constraint symbols of patterns A are either equal or opposite to those of pattern D. Thus, the coefficients spanning the constraint symbols of patterns A, C, and D are tied to only two possibilities (i.e. same sign or alternating) and the result holds.

Finally, letting $s_2 = 1$, it follows that the first m' symbols of pattern B are either all of the same sign or alternating. Applying this result to pattern A and further equating the overlapping symbols between the two patterns, it follows that the first m symbols of pattern A are also either all of the same sign or alternating. The result therefore follows by the previous argument. □

Combining the result of Lemma 3 with that of Theorem 12 yields that an $(n, n-2)$ pattern-eliminating code with $n = L$ is effective over any channel. The result follows letting $2 \leq n \leq L$ and noticing that reducing n does not impair the effectiveness of the code.

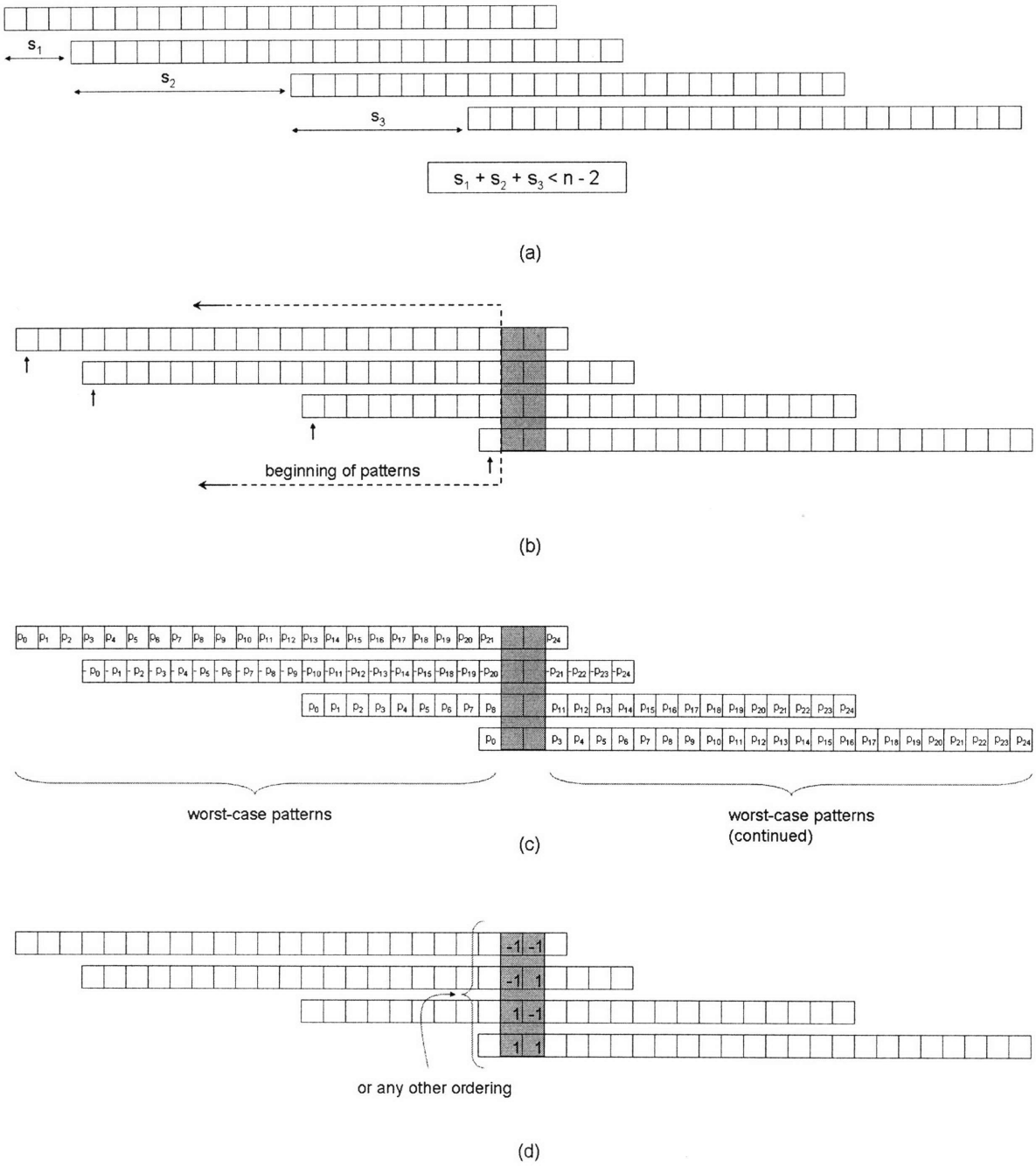

Figure 3-15: Lemma 3 Illustrations

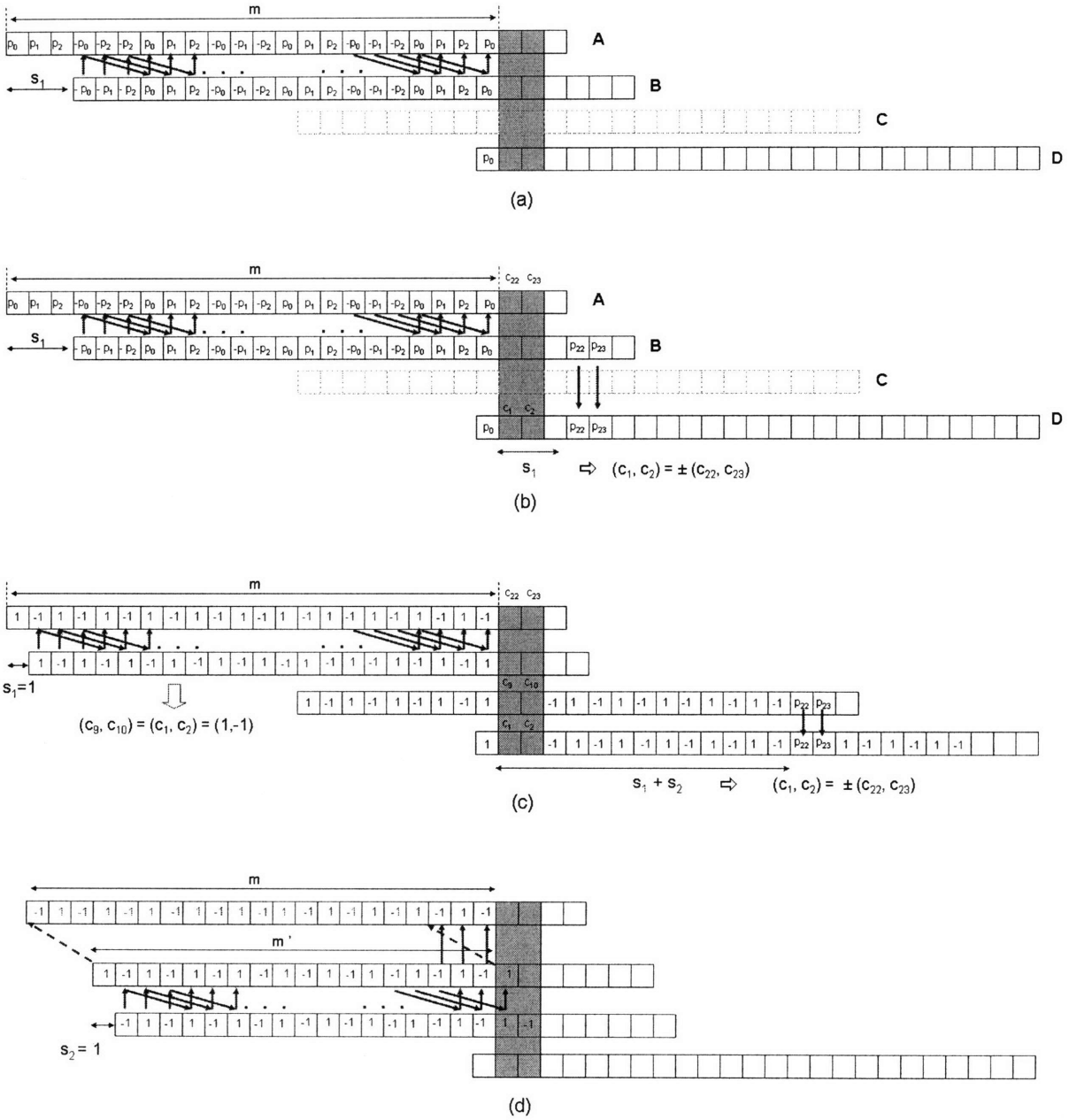

Figure 3-16: Theorem 12 Illustrations

Bibliography

[1] G. D. Forney, *Digital Communications II (6.451) Lecture Notes*, M.I.T. (available at `ocw.mit.edu`).

[2] W. C. Huffman, V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge, U.K.: Cambridge University Press, 2003.

[3] J. G. Proakis, *Digital Communications*, 4^{th} ed., New York, New York: MacGraw-Hill, 2000.

[4] M. Lee, "Channel-and-circuits-aware Energy-efficient Coding for High-speed Links," S.M. thesis, Massachusetts Institute of Technology, 2006.

[5] V. Stojanovic, A. Amirkhany and M.A. Horowitz, "Optimal linear precoding with theoretical and practical data rates in high-speed serial-Link backplane communication," *IEEE International Conference on Communications,*June 2004.

[6] V. Stojanovic, M. Horowitz "Modeling and analysis of high-speed links," *IEEE Custom Integrated Circuits Conference*, pp. 589-594, Sept. 2003.

[7] A. Ho, V. Stojanovic, F. Chen, C. Werner, G. Tsang, E. Alon, R. Kollipara, J. Zerbe, and M. Horowitz, "Common-mode backchannel signaling system for differential high-speed links", in *Symposium on VLSI Circuits. Digest of Technical Papers.*, June 2004, pp. 352-355.

[8] J. Zerbe, "Design, equalization and clock recovery for a 2.5-10gb/s 2-pam/4-pam backplane transceiver cell," IEEE International Solid-State Circuits Conference, February 2003.

[9] J. Ren and M. Greenstreet "A unified optimization framework for equalization filter synthesis," in *Design Automation Conference*, pp. 638-643, 2005.

[10] A. Bessios, W. Stonecypher, A. Agarwal, and J. Zerbe, "Transition-limiting codes for 4-pam signaling in high speed serial links," in *IEEE Global Telecommunications Conference*, December 2003, pp. 3747-3751.

[11] L. E. Thon, H-J. Liaw, "Error-Correction Coding for 10Gb/s Backplane Transmission," *DesignCon* 2004.

[12] D. Carney and E. Chandler, "Error-correction coding in a serial digital multigigabit communication system: Implementation and results," *DesignCon*, 2006.

[13] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, July 1948, pp. 379-423.

[14] A. Khandekar, R. McEliece, and E. Rodemich, "The Discrete Noiseless Channel Revisited," in *Coding, Communications and Broadcasting*, Baldock, U.K.: Research Studies Series Ltd., 2000, pp. 115-137.

[15] C. Pimentel, B. F. Uchoa-Filho, "A Combinatorial Approach to Finding the Capacity of the Discrete Noiseless Channel," in *IEEE Transactions on Information Theory*, vol. 49, no. 8, Aug. 2003, pp.2024-2028.

[16] J. Zerbe, C. Werner, V. Stojanovic, F. Chen, J. Wei, G. Tsang, D. Kim, W. Stonecypher, A. Ho, T. Thrush, R. Kollipara , M. Horowitz, K. Donnelly, "Equalization and Clock Recovery for a 2.5 - 10Gb/s 2-PAM/4-PAM Backplane Transceiver Cell," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2121-2130, Dec. 2003.

[17] IEEE P802.3ap Task Force Channel Model Material (available at `www.ieee802.org/3/ap/public/channel_model`)

[18] A. X. Widmer, P. A. Franaszek, "A DC-Balanced, Partitioned-Block, 8b/ 10b Transmission Code," IBM Journal of Research and Development, vol. 27, no. 5, pp. 440-451, Sept. 1983.

[19] E. W. Weisstein, "Normal Distribution Function." From MathWorld–A Wolfram Web Resource.
http://mathworld.wolfram.com/NormalDistributionFunction.html

[20] H. Kobayashi and D. T. Tang, "Application of Partial-response Channel Coding to Magnetic Recording Systems, IBM Journal of Research and Development, vol. 14, no. 1, 1970.

[21] H. Kobayashi, "Application of probabilistic decoding to digital magnetic recording systems," IBM Journal of Research and Development, vol. 15, pp. 65-74, Jan. 1971.

[22] H. Kobayashi, "Correlative level coding and maximum-likelihood decoding," IEEE Transactions on Information Theory, vol. IT-17, no. 5 , pp. 586-594, Sept. 1971.

[23] G. D. Forney, Jr. "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," IEEE Transactions on Information Theory, vol. IT-18, no. 3, pp. 363-378, May 1972.

[24] G. D. Forney, Jr., "Coset codes I: Geometry and classification," IEEE Transactions on Information Theory, vol. 34, no. 5, pp. 1123-1151, September 1988.

[25] G. D. Forney, Jr., "Coset codes II: Binary lattices and related codes," IEEE Transactions on Information Theory, vol. 34, no. 5, pp. 1152-1187, September 1988.

[26] G. D. Forney, Jr., "The Viterbi Algorithm", Proceedings of the IEEE, pp. 268-278, March 1973.

[27] M. Tomlinson, "New automatic equaliser employing modulo arithmetic,", Electronic Letters, vol. 7, no. 5, pp. 138-139, March 1971.

[28] H. Harashima and H. Miyakawa, "Matched-transmission technique for channels with intersymbol interference," IEEE Transactions on Communications, vol. COM-20, pp. 774-779, August 1972.

[29] J. T. Aslanis, S. Kasturia, G. P. Dudevoir, J. M. Cioffi, "Vector coding for partial response channels," IEEE Military Communications Conference, vol 2. pp. 667-671, San Diego, CA, October 1988.

[30] A. Kavcic, X. Ma, N. Varnica, "Matched information rate codes for partial response channels," IEEE Transactions on Information Theory, vol. 51, no. 3, pp. 973-989 March 2005

[31] H. W. Song, J.F. Liu, B.V.K.V. Kumar, "Large girth cycle codes for partial response channels," IEEE Transactions on Magnetics, vol. 40, no. 4, pp. 3084-3086, July 2004

[32] W. Tan, J. R. Cruz, "Performance evaluation of low-density parity-check codes on partial-response channels using density evolution", IEEE Transactions on Communications, vol. 52, no. 8, pp. 1253-1256, August 2004

[33] B. Vasic, J. Park, E. M. Kurtas, "Soft-decision decoding of Reed-Muller codes with applications to partial response channels,", IEEE Transactions on Magnetics, vol. 40, no. 4, pp. 3120-3122 July 2004

[34] J. Lee, K. Y. Kim, J. J. Lee, et al., "LDPC code decoding adapted to the precoded partial response magnetic recording channels,", Journal of Magnetism and Magnetic Materials vol. 272, pp. 2282-2284, Part 3 Special Issue, May 2004

[35] H. W. Song, B. V. K. V. Kumar, "Low-density parity-check codes for partial response channels,", IEEE Signal Processing Magazine, vol. 21, no. 1, pp. 56-66, January 2004

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

[37] R. Behrens and A. Armstrong, "An advanced ReadiWrite Channel for Magnetic Storage," Proc. 26th Asilomar Conference on Signals, Systems, and Computing, pp. 956-960, Pacific Grove, CA, October, 1992.

[38] J. Moon and B. Brickner, "Maximum transition run codes for data storage systems," IEEE Transactions on Magnetics, vol. 32, no. 5, pp. 3992-3994, September 1996.

[39] K. A. Schouhamer Imink, P. H. Siegel, J. K. Wolf, "Codes for Digital Recorders," IEEE Transactions on Information Theory, vol. 44, no. 6, October 1998.

[40] R. Karabed, P.H. Siegel, E. Soljanin, "Constrained Coding for Binary Channels with High Intersymbol Interference," IEEE Transactions on Information Theory, vol. 45, no. 6, September 1999.

[41] T. L. Poo, B. H. Marcus, "Time-varying Maximum Transition Run Constraints," IEEE Transactions on Information Theory, vol. 52, no. 10, October 2006.

[42] T. L. Poo, P, Chaichanavong, B. H. Marcus, "Tradeoff Functions for Constrained Systems with Unconstrained Positions," IEEE Transactions on Information Theory, vol. 52, no. 4, April 2006.

[43] A. J. van Wijngaarden, K. A. Schouhamer immink, "Maximum Runlength-limited Codes with Error Control Capabilities", IEEE Journal on Selected Areas in Communications, vol. 19, No. 4, April 2001.

[44] H. F. Tsai, P. H. Liu, Y. Y. , Lin, " TMTR codes for partial response channels," The Institute of Electronics, Information and Communication Engineers (IEIC) Transactions on Electronics, vol. E88C, No. 9, pp. 1903-1908, September 2005.

[45] P. Chaichanavong, B. H. Marcus, "Optimal block-type-decodable encoders for constrained systems," IEEE Transactions on Information Theory, vol. 49, no. 5, pp. 1231-1250, May 2003.

[46] J. M. Cioffi, *EE379A course reader*, Stanford University (on-line resource available at `http://www.stanford.edu/class/ee379a/`)

[47] E. R. Kretzmer. "Generalization of a technique for binary data communication." IEEE Transactions on Communication Technology, vol. 14, pp. 67-68, February 1966.

[48] P. J. Smith, M. Shafi, H. Gao, "Quick simulation: a review of importance sampling techniques in communications systems," *IEEE Journal on Selected Areas in Communications,*vol. 15, no. 4, pp. 597-613, 1997.

[49] B. K. Casper, M. Haycock, R. Mooney, "An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes," *IEEE Symposium on VLSI Circuits*, pp. 54-57, June 2002.

[50] B. Ahmad, "Performance Specification of Interconnects," *DesignCon 2003.*

[51] A. Sanders, M. Resso, J. D'Ambrosia "Channel Compliance Testing Utilizing Novel Statistical Eye Methodology," *DesignCon 2004.*

[52] Cody, W. J., "Rational Chebyshev Approximations for the Error Function", Mathematics of Computation, pp. 631-638, 1969.