UCLA
UCLA Electronic Theses and Dissertations

Title
Reliability-output Decoding and Low-latency Variable-length Coding Schemes for Communication with Feedback

Permalink
https://escholarship.org/uc/item/6gw9s7q7

Author
Williamson, Adam

Publication Date
2014

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

University of California
Los Angeles

Reliability-output Decoding and Low-latency Variable-length Coding Schemes for Communication with Feedback

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Adam Royce Williamson

2014

© Copyright by
Adam Royce Williamson
2014

Abstract of the Dissertation

Reliability-output Decoding and Low-latency Variable-length Coding Schemes for Communication with Feedback

by

Adam Royce Williamson

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2014

Professor Richard D. Wesel, Chair

Feedback links are ubiquitous in modern communication systems. This dissertation is focused on the short-blocklength performance of coded communication systems with feedback and is organized in three main parts. The first section presents a novel reliability-output decoding algorithm for tail-biting convolutional codes, based on Raghavan and Baum's Reliability-Output Viterbi Algorithm (ROVA) for terminated convolutional codes. Whereas terminated convolutional codes suffer from a rate penalty at short blocklengths, tail-biting convolutional codes do not suffer from rate loss, making them throughput-efficient and suitable for used in reliability-based retransmission schemes with short blocklengths.

The second portion of the dissertation analyzes the performance of deterministic variable-length feedback coding schemes, focusing on blocklengths less than 300 symbols. In both the decision-feedback and information-feedback settings, we demonstrate that tail-biting convolutional codes can deliver rates surpassing the random-coding lower bound at blocklengths less than 100 symbols. The decision-feedback scheme uses the tail-biting ROVA to determine when to stop transmission, which requires only a single bit of feedback (ACK/NACK) after each decoding attempt. In contrast, the information-feedback scheme employs two-phase

incremental redundancy and uses feedback of the received symbols to confirm or reject the decoder's tentative estimate. Finally, we discuss implications of these schemes when used in practical systems, namely the performance when decoding is limited to packets instead of individual symbols.

Finally, using the information-theoretic framework of the second section, the third part of this dissertation investigates the energy efficiency of variable-length feedback codes compared to that of fixed-length block codes without feedback. While the latency reduction obtained with feedback can decrease transmitter power consumption, attempting decoding after every received symbol significantly increases receiver power consumption. Care must be taken to choose decoding intervals that balance the competing interests of transmitter and receiver power.

The dissertation of Adam Royce Williamson is approved.

Lara Dolecek

Mario Gerla

Greg Pottie

Richard D. Wesel, Committee Chair

University of California, Los Angeles

2014

Table of Contents

List of Figures

List of Tables

Acknowledgments

I am grateful to Professor Richard Wesel for his guidance during my graduate studies at UCLA. His encouragement and enthusiasm helped me to work hard week after week. I appreciate that he fostered an environment of mutual respect within our research group and gave me the autonomy to pursue my own ideas.

I wish to thank the faculty members on my committee, Professors Lara Dolecek, Mario Gerla, and Greg Pottie, for their time and insightful questions. I had many good teachers at UCLA, but these were some of the best. Thanks go as well to Professor Dariush Divsalar, who was willing to answer any question I had, even as a beginning graduate student researcher. Professor Divsalar's recommendation to use the reliability-output Viterbi algorithm played a crucial role in my research trajectory, so I am thankful for his years of experience. I also need to thank the hard-working staff of the Electrical Engineering Department at UCLA, without whom we'd all be lost.

Many fellow students at UCLA have helped me in graduate school, especially my more senior colleagues in the Communication Systems Laboratory, Jiadong Wang, Thomas Courtade, and Tsung-Yi Chen. My years in the group most overlapped with those of Kasra Vakilinia, for which I am grateful. I also had the pleasure of getting to know Matthew Marshall, Sudarsan Ranganathan, Harsha Bhat, Chris Miller, and many other younger students, whose work I look forward to seeing. When I was a new student to the group without much research experience, I especially benefited from close collaboration with Tsung-Yi Chen on my feedback work. This continued to be a fruitful collaboration throughout my time at UCLA.

I am thankful for my coworkers' support of my PhD studies and for their flexibility to allow me to spend time away from work.

Thanks go to my parents for their encouragement throughout all stages of my education. The value they placed on education during my childhood was crucial to my success as a student.

Lastly, I want to thank my wife Hannah for her enduring support. Her confidence in me is unequaled and her encouragement has helped me to complete my PhD. I'm proud to have shared so many parts of our education, both in Rochester and in Los Angeles.

Vita

2008	Bachelor of Science, Electrical and Computer Engineering University of Rochester
2008	Bachelor of Science, Applied Mathematics University of Rochester
2008	Bachelor of Arts, Physics University of Rochester
2011, 2012	Honorable Mention, Graduate Research Fellowship Program National Science Foundation
2012	Master of Science, Electrical Engineering University of California, Los Angeles
2013	PhD Preliminary Exam Award, Electrical Engineering University of California, Los Angeles

Publications

A. R. Williamson, T.-Y. Chen, and R. D. Wesel, "A rate-compatible sphere-packing analysis of feedback coding with limited retransmissions," in *Proc. 2012 IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, MA, USA, July 2012.

A. R. Williamson, T.-Y. Chen, and R. D. Wesel, "Firing the genie: Two-phase short-blocklength convolutional coding with feedback," in *2013 Inf. Theory and Applications Workshop (ITA)*, San Diego, CA, USA, Feb. 2013.

A. R. Williamson, T.-Y. Chen, and R. D. Wesel, "Reliability-based error detection for feedback communication with low latency," in *Proc. 2013 IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, July 2013.

T.-Y. Chen, A. R. Williamson, and R. D. Wesel, "Variable-length coding with feedback: Finite-length codewords and periodic decoding," in *Proc. 2013 IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, July 2013.

T.-Y. Chen, A. R. Williamson, N. Seshadri, and R. D. Wesel, "Feedback communication systems with limitations on incremental redundancy," submitted for publication. Available: http://arxiv.org/abs/1309.0707.

A. R. Williamson, M. J. Marshall, and R. D. Wesel, "Reliability-output decoding of tail-biting convolutional codes," *IEEE Trans. Commun.*, to be published.

T.-Y. Chen, A. R. Williamson, and R. D. Wesel, "Increasing flash memory lifetime by dynamic voltage allocation for constant mutual information," in *2014 Inf. Theory and Applications Workshop (ITA)*, San Diego, CA, USA, Feb. 2014.

T.-Y. Chen, A. R. Williamson, and R. D. Wesel, "Asymptotic expansion and error exponent of two-phase variable-length coding with feedback for discrete memoryless channels," in *Proc. 2014 IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, July 2014.

K. Vakilinia, T.-Y. Chen, S. V. S. Ranganathan, A. R. Williamson, D. Divsalar, and R. D. Wesel, "Short-blocklength non-binary LDPC codes with feedback-dependent incremental transmissions," in *Proc. 2014 IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, July 2014.

CHAPTER 1

Introduction

Feedback links are ubiquitous in modern communication systems and have been studied in the information theory community for decades. Despite the fact that noiseless feedback does not increase the asymptotic (Shannon) capacity of point-to-point, memoryless channels, there are many other benefits of feedback. For example, feedback can simplify encoding and decoding operations and can improve the error exponent, which governs the error probability as a function of blocklength.

Recent developments in finite-blocklength information theory have renewed interest in the benefits of feedback at short blocklengths. Compared to fixed-length block codes without feedback, variable-length coding with feedback can dramatically improve the maximum non-asymptotic rate. This dissertation builds on the recent information theory to demonstrate deterministic coding schemes that perform well at average blocklengths less than 300 symbols. In particular, we show that convolutional codes, which are known to have excellent error performance at short blocklengths, can deliver rates above the information-theoretic random-coding lower bound. In contrast, fixed-length codes without feedback have typically required thousands of bits to achieve rates approaching capacity. Low-latency communication is desirable for many emerging applications, so it is important to investigate coding schemes that provide rates close to capacity without requiring thousand-bit latencies. Machine-to-machine communication, for example, may have packets lengths of only a few bytes.

1.1 Outline of Dissertation

The dissertation is organized in three main parts. The first part (Ch. 2), presents novel reliability-output decoding algorithms for tail-biting convolutional codes, based on Raghavan and Baum's Reliability-Output Viterbi Algorithm (ROVA) for terminated convolutional codes. Terminated convolutional codes start and end in a known state (usually the all-zeros state), which results in a rate penalty due to the fixed termination symbols that must be transmitted in order to arrive at the termination state. This rate penalty can be severe at short blocklengths (e.g., under 100 symbols).

In contrast, tail-biting convolutional codes can start in any state, but must end in the same state. This starting/ending state is unknown at the receiver, which increases the decoding complexity compared to terminated convolutional codes. Tail-biting convolutional do not suffer the rate loss of terminated convolutional codes, making them throughput-efficient and suitable for use in hybrid Automatic Repeat reQuest (hybrid ARQ) systems. However, the ROVA for terminated codes does not accommodate tail-biting codes, which is the motivation for this work.

The tail-biting ROVA developed in this dissertation computes the exact word-error probability of decoded tail-biting codewords by first calculating the posterior probability of the codeword's starting state. Due to the high computational complexity of the tail-biting Viterbi algorithm, several reduced-complexity versions of the tail-biting ROVA are discussed. One approach employs a state-estimation algorithm that selects the maximum a posteriori state based on the posterior distribution of the starting states. Another approach is an approximation to the exact tail-biting ROVA that estimates the word-error probability. A comparison of the computational complexity of each approach is discussed in detail. These reliability-output algorithms apply to both feedforward and feedback tail-biting convolutional encoders.

The second portion of the dissertation (Ch. 3) compares the performance of two categories of variable-length coding schemes, decision feedback and information feedback, in the short-blocklength regime. In both the decision-feedback and information-feedback settings, we demonstrate that tail-biting convolutional codes can deliver rates surpassing the random-coding lower bound at blocklengths less than 100 symbols. The decision-feedback scheme uses the tail-biting ROVA to determine when to stop transmission, which requires only a single bit of feedback (ACK/NACK) after each decoding attempt. In contrast, the information-feedback scheme employs two-phase incremental redundancy and uses feedback of the received symbols to confirm or reject the decoder's tentative estimate. Finally, we discuss the implications of these schemes when used in practical systems. In particular, we highlight the tension between decision feedback and information feedback, the latter of which generally requires a higher-rate feedback link in practice.

In addition to demonstrating explicit variable-length coding schemes, Ch. 3 provides several lower bounds on the maximum rate at short blocklengths, including two improved achievability results based on information feedback. Other theorems in Ch. 3 extend the random-coding lower bound to accommodate variable-length codes formed by repeating length-N block codes, and to accommodate decoding after packets instead of decoding after every individual received symbol.

Using the information-theoretic framework of Ch. 3, the third part of this dissertation (Ch. 4) investigates the energy efficiency of variable-length feedback codes compared to fixed-length block codes without feedback. This chapter demonstrates that while the latency reduction obtained with feedback can decrease transmitter power consumption, attempting decoding after every received symbol significantly increases receiver power consumption. Care must be taken to choose decoding intervals that balance the competing interests of transmitter and receiver power. Finally, Ch. 5 concludes the dissertation.

1.2 Summary of Contributions

The main contributions of this dissertation are as follows:

- This dissertation introduces the following reliability-output decoding algorithms for tail-biting convolutional codes:
  - The straightforward Tail-Biting ROVA (TB ROVA), which decodes the convolutional code and computes the exact posterior probability of the decoded word.
  - The Approximate Tail-Biting ROVA (Approx. TB ROVA), which decodes the convolutional code and estimates (approximates) the posterior probability of the decoded word.
  - The Tail-Biting State-Estimation Algorithm (TB SEA), which determines the maximum a posteriori (MAP) starting state and computes the exact posterior probability of the MAP state. Following TB SEA, the terminated ROVA can be used to decode the terminated code starting and ending in the MAP state.
  - The Post-decoding Reliability Computation (PRC), which computes the exact posterior probability of a decoded word that is supplied to the PRC.
- Using the variable-length feedback (VLF) coding framework, this dissertation presents new achievability results for the finite-blocklength regime:
  - Random-coding lower bounds for repeat-after-N codes and for m-transmission repeat-after-N_m codes provide a theoretical benchmark for evaluating the performance of VLF codes that have limitations on the decoding intervals. Both bounds use random coding with decision feedback.
  - Two-phase information-feedback lower bounds demonstrate how a communication-phase random code followed by a confirmation phase can deliver rates surpassing the random-coding lower bound, which is based on decision feedback. The first

of the two-phase lower bounds uses a fixed-length block code in the confirmation block. The second bound uses a variable-length repetition code in the confirmation phase.

- Explicit constructions using convolutional codes demonstrate deterministic VLF codes that exceed the random-coding lower bound:
  - In the decision-feedback setting, reliability-based retransmission using the TB ROVA provides high rates at short blocklengths. In contrast to retransmission based on Cyclic Redundancy Checks (CRCs), the reliability-based retransmission approach avoids the rate penalty that would be incurred when transmitting additional error-detection symbols.
  - In the information-feedback setting, a two-phase incremental redundancy scheme enables low error rates by confirming the receiver's tentative decoding estimates.
  - For both the two-phase scheme and an incremental redundancy scheme based on the TB ROVA, a blocklength-selection algorithm identifies the set of transmission lengths corresponding to the highest rates at short blocklengths.
- This dissertation proposes a new framework for comparing the energy efficiency of VLF codes and fixed-length block codes with finite message sets:
  - An energy model that includes contributions from both the transmitter and receiver facilitates evaluations of the energy associated with repeated decoding attempts.
  - Numerical examples demonstrate that decoding after every symbol is not optimal, but that with carefully selected decoding intervals, VLF codes offer lower energy consumption than fixed-length block codes.

1.3 Background: Feedback in Modern Communication Systems

Despite the well-known information-theoretic benefits of feedback, some aspects of feedback coding have not been fully utilized in current systems. In this section, we give a brief background on some of the ways feedback is used in modern communication-systems and identify opportunities to take advantage of feedback in future systems.

One of the simplest forms of feedback is ARQ, which has been used extensively in wireline systems for decades, originally with uncoded data packets. Eventually forward error correction (FEC) was added, forming the basis for hybrid ARQ protocols, which divide the responsibility of error control among the forward link (via FEC) and the feedback link (via retransmission requests). The simplest form of hybrid ARQ, type I, refers to a single FEC codeword that is retransmitted until the receiver sends an acknowledgment. In type-II hybrid ARQ, also called incremental redundancy, retransmissions do not repeat the original codeword but instead consist of additional parity symbols that the receiver can use to decode. ARQ and hybrid ARQ are examples of decision-feedback schemes and only require 1 bit of feedback per forward packet. Typically the receiver's decision is based on a Cyclic Redundancy Check (CRC), which provides error detection but reduces the information rate.

Hybrid ARQ is especially beneficial for wireless channels and continues to receive attention, both in the information theory literature (e.g., [5–7]) and in industry standards such as 3GPP (e.g., [8, 9]). Despite the wide body of research indicating the throughput benefits of incremental redundancy, many modern communication systems include type-I hybrid ARQ protocols but do not incorporate incremental redundancy (i.e., attempting decoding with progressively increasing blocklengths). The IEEE 802.11 wireless standard, for example, does not allow for incremental redundancy at the physical layer.

One reason that system architects have avoided incremental redundancy in the past is increased system complexity. As compared to fixed-length-coding schemes, incremental redundancy schemes incur additional complexity both in their design and their operation.

Additional signaling in packet overheads must be used to indicate which retransmission a particular received block belongs to, and larger receive buffers are required to store previously received blocks. These complexity concerns were noted in a 2001 comparison of non-adaptive incremental redundancy and Chase combining, which suggested that although incremental redundancy delivered superior throughput in most cases, the additional complexity was not worthwhile [10]. However, more recent research exploring adaptive incremental redundancy schemes in 3GPP [9,11,12] has claimed that the benefits of incremental redundancy outweigh the complexity concerns for next-generation systems.

Perhaps as a result of these insights, greater attention has been given to incremental redundancy in the communication literature in the past decade (e.g., [13–15]). For example, Visotsky et al. [13] analyze the throughput and latency of hybrid ARQ schemes in the same manner that we have in Ch. 3, though [13] is not focused on short-blocklength performance. Recognizing the rate penalty incurred by transmitting additional parity bits for CRCs, Fricke et al. [15] demonstrated the throughput benefits of a reliability-based retransmission scheme. Fricke et al. [15] investigated the performance of convolutional codes in the type-I hybrid ARQ setting, but wasn't focused on short blocklengths. The reliability-based retransmission scenario in [15] provides the basis for the tail-biting ROVA in Ch. 2.

CHAPTER 2

Reliability-output Decoding of Tail-biting Codes

2.1 Introduction

Raghavan and Baum's reliability-output Viterbi algorithm (ROVA) [1] uses the sequence-estimation property of the Viterbi algorithm to calculate the exact word-error probability of a received convolutional code sequence. In general, the ROVA can be used to compute the word-error probability for any finite-state Markov process observed via memoryless channels (i.e., processes with a trellis structure). However, the ROVA is only valid for processes that terminate in a known state (usually the all-zeros state). For codes with large constraint lengths, a significant rate penalty is incurred due to the additional symbols that must be transmitted in order to arrive at the termination state.

Tail-biting convolutional codes can start in any state, but must terminate in the same state. The starting/terminating state is unknown at the receiver. These codes do not suffer the rate loss of terminated codes, making them throughput-efficient (see, e.g., [16] and [17, Ch. 12.7]). The tail-biting technique is commonly used for short-blocklength coding.

2.1.1 Overview and Contributions

In this chapter, we extend the ROVA to compute the word-error probability for tail-biting codes. First, we present a straightforward approach, which we call the tail-biting ROVA (TB ROVA). TB ROVA invokes the original ROVA for each of the possible starting states s. The complexity of this straightforward approach is large, proportional to $2^{2\nu}$ for standard binary convolutional codes (and $q^{2\nu}$ for convolutional codes over Galois field $GF(q)$), where ν is the number of memory elements in a minimal encoder.

We explore several approaches to reduce the complexity of TB ROVA. We first introduce a post-decoding algorithm that computes the reliability of codewords that have already been decoded by an existing tail-biting decoder, including possibly suboptimal decoders. We then propose a new tail-biting decoder that uses the posterior distribution of the starting states to identify the most probable starting state of the received sequence. Finally, we discuss how to use Fricke and Hoeher's simplified (approximate) ROVA [18] for each of the q^{ν} initial states, which reduces the complexity of the word-error probability computation.

The reliability-output algorithms presented in this chapter apply to both feedforward (non-recursive) and feedback (recursive) convolutional encoders. However, as pointed out by Ståhl et al. [19], it is not possible to have a one-to-one mapping from information words to codewords and still fulfill the tail-biting restriction for feedback encoders at certain tail-biting codeword lengths. Ståhl et al. [19] provide conditions for when tail-biting will work for recursive encoders and also describe how to determine the starting state corresponding to an input sequence. In the cases where the tail-biting technique works for feedback encoders, there is a one-to-one mapping between input sequences and codewords, and the reliability-output algorithms in this chapter are valid.

The remainder of this chapter proceeds as follows: Sec. 2.1.2 reviews the related literature and Sec. 2.1.3 introduces notation. Sec. 2.2 reviews Raghavan and Baum's ROVA and discusses how to extend it to tail-biting codes. The simplified ROVA for tail-biting codes is discussed in Sec. 2.3. Sec. 2.4 presents the Post-decoding Reliability Computation (PRC) for tail-biting codes and Sec. 2.5 introduces the Tail-Biting State-Estimation Algorithm (TB SEA). Sec. 2.5.4 discusses an alternative to TB SEA using the tail-biting BCJR algorithm. Sec. 2.6 evaluates the complexity of the proposed algorithms, and Sec. 2.7 shows numerical examples of the computed word-error probability and the actual word-error probability. Sec. 2.8 concludes the chapter.

2.1.2 Related Literature

There are a number of reliability-based decoders for terminated convolutional codes, most notably the Yamamoto-Itoh algorithm [20], which computes a reliability measure for the decoded word, but not the exact word-error probability. In [15], Fricke and Hoeher use Raghavan and Baum's ROVA in a reliability-based type-I hybrid Automatic Repeat reQuest (ARQ) scheme.

Hof et al. [21] modify the Viterbi algorithm to permit generalized decoding according to Forney's generalized decoding rule [22]. When the generalized decoding threshold is chosen for maximum likelihood (ML) decoding with erasures and the erasure threshold is chosen appropriately, this augmented Viterbi decoder is equivalent to the ROVA.

A type-II hybrid ARQ scheme for incremental redundancy using punctured terminated convolutional codes is presented by Williamson et al. [23]. In [23], additional coded symbols are requested when the word-error probability computed by the ROVA exceeds a target word-error probability. This word-error requirement facilitates comparisons with recent work in the information theory community [24, 25]. Polyanskiy et al. [25] investigate the maximum rate achievable at short blocklengths with variable-length feedback codes. While [23] shows that terminated convolutional codes can deliver throughput above the random-coding lower bound of [25], the rate loss from termination is still significant at short blocklengths. To avoid the termination overhead, it is imperative to have a reliability-output decoding algorithm for tail-biting codes.

In contrast to the decoding algorithms for terminated codes, Anderson and Hladik [26] present a tail-biting maximum a posteriori (MAP) decoding algorithm. This extension of the BCJR algorithm [27] can be applied to tail-biting codes with a priori unequal source data probabilities. As with the BCJR algorithm, [26] computes the posterior probabilities of individual data symbols. In contrast, the ROVA [1] and the tail-biting reliability-based decoders in this chapter compute the posterior probabilities of the codeword.

More importantly, the tail-biting BCJR of [26] is only an approximate symbol-by-symbol MAP decoder, as pointed out in [28] and [29]. Because the tail-biting restriction is not strictly enforced, non-tail-biting "pseudocodewords" can cause bit errors, especially when the ratio of the tail-biting length L to the memory order M is small (i.e., $L/M \approx$ 1-2). Further comparisons with the tail-biting BCJR are given in Sec. 2.5.4. An exact symbol-by-symbol MAP decoder for tail-biting codes is given in [30, Ch. 7].

Handlery et al. [31] introduce a suboptimal, two-phase decoding scheme for tail-biting codes that computes the approximate posterior probabilities of each starting state and then uses the standard BCJR algorithm to compute the posterior probabilities of the source symbols. This approach is compared to the tail-biting BCJR of [26] and exact MAP decoding in terms of bit-error-rate (BER) performance. Both the two-phase approach of [31] and the tail-biting BCJR of [26] perform close to exact MAP decoding when L/M is large, but suffer a BER performance loss when L/M is small.

Although it does not compute the word-error probability, Yu [32] introduces a method of estimating the initial state of tail-biting codes, which consists of computing a pre-metric for each state based on the last M observations of the received word. This pre-metric is then used to initialize the path metrics of the main tail-biting decoder (e.g., the circular Viterbi decoder [33]), instead of assuming that all states are equally likely at initialization. The state-estimation method of [32], which is not maximum-likelihood, is limited to systematic codes and a special configuration of nonsystematic codes that allows information symbols to be recovered from noisy observations of coded symbols.

Because tail-biting codes can be viewed as circular processes [16, 33], decoding can start at any symbol. Wu et al. [34] describe a reliability-based decoding method that compares the log likelihood-ratios of the received symbols in order to determine the most reliable starting-location for tail-biting decoders. Selecting a reliability-based starting location reduces the error probability by minimizing the chance of choosing non-tail-biting paths early in the decoding process. Wu et al. [34] apply this approach to existing suboptimal decoders, including the wrap-around Viterbi algorithm of [35]. As with [32], [34] does not compute the word-error probability.

Pai et al. [36] generalizes the Yamamoto-Itoh algorithm to handle tail-biting codes and uses the computed reliability measure as the retransmission criteria for hybrid ARQ. When there is a strict constraint on the word-error probability, however, this type of reliability measure is not sufficient to guarantee a particular undetected-error probability. Providing such a guarantee motivates the word-error probability calculations in this chapter (instead of bit-error probability as in [26, 28–32]).

2.1.3 Notation

We use the following notation in this chapter: $\mathrm{P}(X = x)$ denotes the probability mass function (p.m.f.) of discrete-valued random variable X at value x, which we also write as $\mathrm{P}(x)$. The probability density function (p.d.f.) of a continuous-valued random variable Y at value y is $\mathrm{f}(Y = y)$, sometimes written as $\mathrm{f}(y)$. In general, capital letters denote random variables and lowercase letters denote their realizations. Letters with superscripts denote vectors, as in $y^\ell = (y_1, y_2, \ldots, y_\ell)$, while subscripts denote a particular element of a vector: y_i is the ith element of y^ℓ. We use the hat symbol to denote the output of a decoder, e.g., $\hat{x}$ is the codeword chosen by the Viterbi algorithm.

2.2 The Reliability-output Viterbi Algorithm

Raghavan and Baum's reliability-output Viterbi algorithm [1] augments the canonical Viterbi decoder with the computation of the word-error probability of the maximum-likelihood (ML) codeword. In this section, we provide an overview of the ROVA.

For rate-k/n convolutional codes with L trellis segments and input alphabet q, we denote the ML codeword as $\hat{x}^L = \hat{x}$ and the noisy received sequence as $y^L = y$. The probability that the ML decision is correct given the received word is $\mathrm{P}(X = \hat{x}|Y = y) = \mathrm{P}(\hat{x}|y)$, and the word-error probability is $\mathrm{P}(X \neq \hat{x}|Y=y) = 1 - \mathrm{P}(\hat{x}|y)$. The probability of successfully decoding can be expressed as follows:

$$\mathrm{P}(\hat{x}|y) = \frac{\mathrm{f}(y|\hat{x})\mathrm{P}(\hat{x})}{\mathrm{f}(y)} = \frac{\mathrm{f}(y|\hat{x})\mathrm{P}(\hat{x})}{\sum_{x'} \mathrm{f}(y|x')\mathrm{P}(x')}, \tag{2.1}$$

where we have used $\mathrm{f}(y|\hat{x})$ to denote the conditional p.d.f. of the real-output channel (e.g., the binary-input AWGN channel). This may be replaced by the conditional p.m.f. $\mathrm{P}(y|\hat{x})$ for discrete-output channels (e.g., the binary symmetric channel).

The probability of correctly decoding can be further simplified if each of the codewords x' is a priori equally likely, i.e., $\mathrm{P}(x') = \mathrm{P}(\hat{x})\ \forall\ x'$, which we assume for the remainder of the chapter. This assumption yields

$$\mathrm{P}(\hat{x}|y) = \frac{\mathrm{f}(y|\hat{x})}{\sum_{x'} \mathrm{f}(y|x')}. \tag{2.2}$$

In general, the denominator in (2.2) may be computationally intractable when the message set cardinality is large. However, the ROVA [1] takes advantage of the trellis structure of convolutional codes to compute $\mathrm{P}(\hat{x}|y)$ exactly with complexity that is linear in the block-length and exponential in ν (i.e., it has complexity on the same order as that of the original Viterbi algorithm). This probability can also be computed approximately by the simplified (approximate) ROVA [18], which will be discussed further in Sec. 2.3.

The ROVA can compute the probability of word error for any finite-state Markov process observed via memoryless channels (e.g., in maximum-likelihood sequence estimation for signal processing applications). In the remainder of this chapter, we use the example of convolutional encoding and decoding, but the ROVA and our tail-biting trellis algorithms apply to any finite-state Markov process.

2.2.1 Conditioning on the Initial State

Raghavan and Baum's ROVA applies only to codes that begin and end at a known state. Each of the probabilities $\mathrm{f}(y|x')$ in (2.2) is implicitly conditioned on the event that the receiver knows the initial and final state of the convolutional encoder.

To be precise, ROVA beginning and ending at the same state s, which we shall denote as ROVA(s), effectively computes the following:

$$\underbrace{\mathrm{P}(\hat{x}_s|y,s)}_{\text{computed by ROVA}(s)} = \frac{\mathrm{f}(y|\hat{x}_s,s)\mathrm{P}(\hat{x}_s|s)}{\mathrm{f}(y|s)} = \frac{\mathrm{f}(y|\hat{x}_s,s)}{\sum\limits_{x'_s}\mathrm{f}(y|x'_s,s)}, \tag{2.3}$$

where the limit x'_s in the summation of the denominator denotes that the enumeration for the summation is over all codewords x'_s with starting state s, and $\mathrm{f}(y|x'_s,s)$ is shorthand for $\mathrm{f}(Y=y|X=x'_s,S=s)$. In summary, ROVA($s$) computes the ML codeword $\hat{x}_s$ corresponding to starting state s, the posterior probability of that codeword given s, $\mathrm{P}(\hat{x}_s|y,s)$, and the probability of the received sequence given s, $\mathrm{f}(y|s)$. The inputs and outputs of ROVA(s) are illustrated in the block diagram of Fig. 2.1.

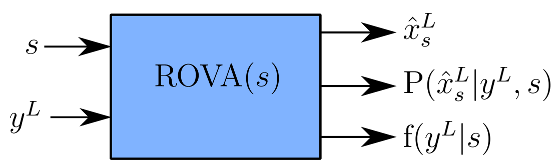


Figure 2.1: Block diagram of Raghavan and Baum's ROVA(s) [1].

For tail-biting codes, we are interested in computing the quantity $\mathrm{P}(\hat{x}|y)$ without conditioning on the unknown starting and ending state s:

$$\mathrm{P}(\hat{x}|y) = \sum_s \mathrm{P}(\hat{x}|y,s)\mathrm{P}(s|y). \tag{2.4}$$

The ML codeword $\hat{x}$ has an associated initial state, $\hat{s}$. Note that $\mathrm{P}(\hat{x}|y,s) = 0$ unless $s = \hat{s}$, since $\hat{x}$ is not a possible codeword for any starting state other than $\hat{s}$. Thus, we have:

$$\mathrm{P}(\hat{x}|y) = \mathrm{P}(\hat{x}|y,\hat{s})\mathrm{P}(\hat{s}|y). \tag{2.5}$$

Thus, the tail-biting ROVA (TB ROVA) must compute the probability $\mathrm{P}(\hat{x}|y)$ of successful decoding in (2.5) by weighting $\mathrm{P}(\hat{x}|y,\hat{s})$ with $\mathrm{P}(\hat{s}|y)$. (For the original ROVA with a known starting state $\hat{s}$, $\mathrm{P}(\hat{s}|y) = 1$ and $\mathrm{P}(s'|y) = 0\ \forall\ s' \neq \hat{s}$.)

Using the fact that each of the initial states s is equally likely *a* priori (i.e., $\mathrm{P}(s) = \mathrm{P}(s')\ \forall s \neq s'$), we have:

$$\mathrm{P}(\hat{s}|y) = \frac{\mathrm{f}(y|\hat{s})}{\sum_{s'} \mathrm{f}(y|s')}. \tag{2.6}$$

This finally yields

$$\underbrace{\mathrm{P}(\hat{x}|y)}_{\text{computed by TB ROVA}} = \frac{\overbrace{\mathrm{P}(\hat{x}|y,\hat{s})\mathrm{f}(y|\hat{s})}^{\text{computed by ROVA}(\hat{s})}}{\sum_{s'} \underbrace{\mathrm{f}(y|s')}_{\text{computed by ROVA}(s')}}, \tag{2.7}$$

where the summation in the denominator of (2.7) is over all q^ν possible initial states.

2.2.2 A Straighforward Tail-biting ROVA

A straightforward ML approach to decoding tail-biting codes is to perform the Viterbi algorithm VA(s), for each possible starting state $s = 0, 1, \ldots, q^\nu - 1$. The ML codeword $\hat{x}$ is then chosen by determining the starting state with the greatest path metric (i.e., the greatest probability). As shown in Fig. 2.2, this approach will work for the ROVA as well: perform ROVA(s) for each possible s and then pick $\hat{x}$ and its starting state $\hat{s}$. The probability $\mathrm{P}(\hat{x}|y)$ is then computed as in (2.7), using $\mathrm{P}(\hat{x}|y,\hat{s})$ from the ROVA for the ML starting state and the $\mathrm{f}(y|s)$ terms produced by the ROVAs for all the states. This approach is illustrated in the block diagram of Fig. 2.2.

2.3 The Simplified ROVA for Tail-biting Codes

This section proposes replacing the exact word-error computations of Sec. 2.2.2's straightforward TB ROVA with an estimated word-error probability, using Fricke and Hoeher's simplified (approximate) ROVA [18]. This approach requires a Viterbi decoder for each starting state to select the ML codeword for that state. Fricke and Hoeher's [18] simplified ROVA for starting state s, which we call Approx ROVA(s), estimates the probability $\mathrm{P}(x_s|y, s)$.

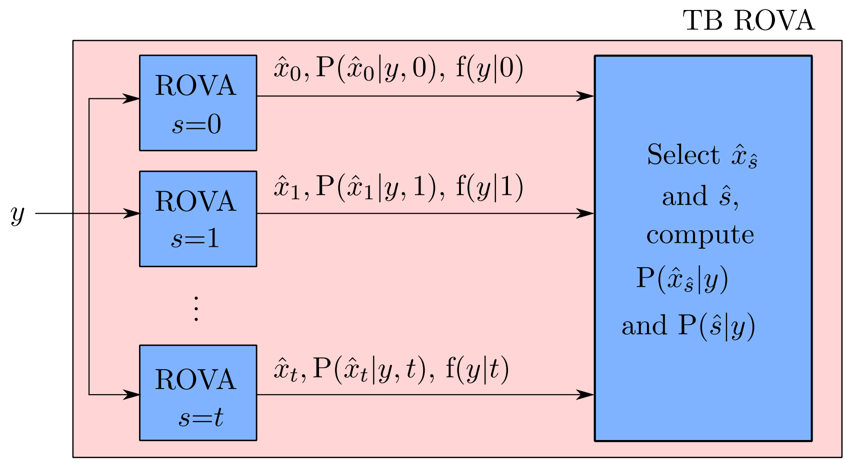


Figure 2.2: Block diagram of the straightforward tail-biting ROVA (TB ROVA), which performs the ROVA(s) for each possible starting state s. The largest possible state is $t = q^\nu - 1$.

Substituting this estimate $\tilde{\mathrm{P}}(x_s|y,s)$ into (2.7), we have the following approximation:

$$\tilde{\mathrm{P}}(\hat{x}|y) = \frac{\overbrace{\tilde{\mathrm{P}}(\hat{x}|y,\hat{s})}^{\text{computed by Approx ROVA}(\hat{s})} \qquad \mathrm{f}(y|\hat{s})}{\sum_s \mathrm{f}(y|s)}. \tag{2.8}$$

While $\mathrm{f}(y|s)$ is not computed directly by Approx ROVA(s), we can approximate it with quantities available from Approx ROVA(s) as

$$\begin{aligned} \mathrm{f}(y|s) \approx \tilde{\mathrm{f}}(y|s) &= \frac{\mathrm{f}(y|x_s,s)\mathrm{P}(x_s|s)}{\tilde{\mathrm{P}}(x_s|y,s)} && (2.9)\\ &= \frac{\mathrm{f}(y|x_s,s)}{q^{kK}\tilde{\mathrm{P}}(x_s|y,s)}, && (2.10)\end{aligned}$$

when all q^{kK} codewords with starting state s are equally likely, where K is the number of trellis segments before termination. Note $\mathrm{f}(y|x_s,s)$ can be calculated by the Viterbi algorithm for starting state s.

Equations (2.8) and (2.10) lead to the following estimate of the word-correct probability:

$$\tilde{\mathrm{P}}(\hat{x}|y) \approx \frac{\overbrace{\tilde{\mathrm{P}}(\hat{x}|y,\hat{s})\tilde{\mathrm{f}}(y|\hat{s})}^{\text{computed by Approx ROVA}(\hat{s})}}{\sum_s \underbrace{\tilde{\mathrm{f}}(y|s)}_{\text{computed by Approx ROVA}(s)}}. \tag{2.11}$$

We refer to the overall computation of $\tilde{\mathrm{P}}(\hat{x}|y)$ in (2.11) as Approx TB ROVA. Sec. 2.6 provides a discussion of its complexity and Sec. 2.7 presents simulation results.

Note that despite the approximations, the simplified ROVA chooses the ML codeword for terminated codes. For the tail-biting version Approx TB ROVA, as long as the winning path metric of each starting/ending state is used to determine the ML state $\hat{s}$, the decoded codeword will also be ML (and the same as the codeword chosen by the exact tail-biting ROVA in Sec. 2.2.2). However, if the approximate reliabilities $\tilde{\mathrm{P}}(x_s|y,s)$ are used instead of the path metrics to select the decoded word $\hat{x}$ as $\hat{x} = \arg\max\limits_{s} \tilde{\mathrm{P}}(x_s|y,s)$, it is possible that the decoded word will not be ML (if the channel is noisy enough).

2.4 Post-decoding Reliability Computation

There are q^ν possible starting states that must be evaluated in the straightforward TB ROVA of Sec. 2.2.2 and Fig. 2.2. Thus it may be beneficial to instead use an existing reduced-complexity tail-biting decoder to find $\hat{x}$, and then compute the reliability separately. Many reduced-complexity tail-biting decoders take advantage of the circular decoding property of tail-biting codes. Some of these approaches are not maximum likelihood, such as the wrap-around Viterbi algorithm or Bidirectional Viterbi Algorithm (BVA), both discussed in [35]. The A* algorithm [37–39] is one ML alternative to the tail-biting decoding method described in Sec. 2.2.2. Its complexity depends on the SNR.

Suppose that a decoder has already been used to determine $\hat{x}$ and its starting state $\hat{s}$, and that we would like to determine $\mathrm{P}(\hat{x}|y)$. One operation of ROVA$(\hat{s})$ would compute the probability $\mathrm{P}(\hat{x}|y,\hat{s})$, but the probability $\mathrm{P}(\hat{s}|y)$ required by (2.5) would still be undetermined. Must we perform ROVA(s) for each $s \neq \hat{s}$ in order to compute $\mathrm{P}(\hat{s}|y)$ as in (2.6)? This section shows how to avoid this by combining the computations of the straightforward approach into a novel Post-decoding Reliability Computation (PRC) for tail-biting codes.

Fig. 2.3 shows a block diagram of PRC. For a rate-k/n tail-biting convolutional code with

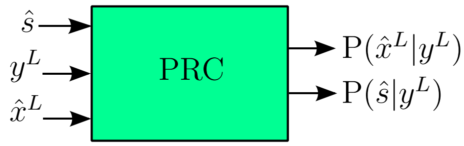


Figure 2.3: Block diagram of the Post-decoding Reliability Computation (PRC).

ν memory elements, PRC takes the following inputs: a received sequence y^L with L trellis segments, a *candidate codeword* $\hat{x}^L$ corresponding to a *candidate path* in the trellis, and the starting/ending state $\hat{s}$ of the candidate codeword. The goal is to compute the posterior probability of the candidate codeword, $\mathrm{P}(\hat{x}^L|y^L)$. The candidate codeword selected by the decoder may not be the ML codeword. PRC computes its true reliability regardless.

Raghavan and Baum's ROVA [1] performs the traditional add-compare-select operations of the Viterbi algorithm and then computes, for every state in each trellis segment, the posterior probability that the survivor path is correct and the posterior probability that one of the non-surviving paths at the state is correct. Upon reaching the end of the trellis (the Lth segment), having selected survivor paths at each state, there will be one survivor path corresponding to the ML codeword.

In contrast, with the candidate path already identified, PRC processes the trellis without explicitly selecting survivors. PRC computes the reliability of the candidate path and the overall reliability of all other paths.

2.4.1 PRC Overview

We define the following events at trellis stage ℓ ($\ell \in \{1, 2, \ldots, L\}$):

- $p_\ell^{\hat{s}\to j} = \{$the candidate path from its beginning at state $\hat{s}$ to its arrival at state j in segment ℓ is correct$\}$
- $\bar{p}_\ell^{s\to r} = \{$some path from its beginning at state s to its arrival at state r in segment ℓ is correct (including possibly the candidate path if $\hat{s} = s$) $\}$

- $b_\ell^{i\to j}$ = {the branch from state i to state j at time ℓ is correct}

For $\nu = 3$ memory elements, Fig. 2.4 gives some examples of the paths corresponding to each of these events. The black branches in Fig. 2.4 constitute all of the paths in the event $\bar{p}_7^{s\to r}$. The red branches in Fig. 2.4 show the candidate path corresponding to the event $p_7^{\hat{s}\to j}$. The posterior probability that the red candidate path starting at state $\hat{s}$ is correct is $\mathrm{P}(p_7^{\hat{s}\to j}|y^7)$. The posterior probability that any of the paths that started at state s and arrive at state r in segment 7 are correct is $\mathrm{P}(\bar{p}_7^{r\to s}|y^7)$. Note that since some branch transitions are invalid in the trellis, $\mathrm{P}(p_\ell^{\hat{s}\to j})$ and $\mathrm{P}(b_\ell^{i\to j})$ may be zero for invalid states and branches in segment ℓ.

The path-correct probabilities $\mathrm{P}(p_\ell^{\hat{s}\to j})$ and $\mathrm{P}(\bar{p}_\ell^{s\to r})$ can be expressed recursively in terms of the probabilities of the previous trellis segments' paths being correct. Conditioned on the noisy channel observations $y^\ell = (y_1, y_2, \ldots, y_\ell)$, the path-correct probability for the candidate path (which passes through state i in segment $\ell - 1$) is

$$\begin{aligned}
\mathrm{P}(p_\ell^{\hat{s}\to j}|y^\ell) &= \mathrm{P}(p_{\ell-1}^{\hat{s}\to i}, b_\ell^{i\to j}|y^\ell) && (2.12)\\
&= \mathrm{P}(b_\ell^{i\to j}|y^\ell, p_{\ell-1}^{\hat{s}\to i})\mathrm{P}(p_{\ell-1}^{\hat{s}\to i}|y^\ell) && (2.13)\\
&= \frac{\mathrm{f}(y_\ell|y^{\ell-1}, p_{\ell-1}^{\hat{s}\to i}, b_\ell^{i\to j})}{\mathrm{f}(y_\ell|y^{\ell-1})} && (2.14)\\
&\quad \times \mathrm{P}(b_\ell^{i\to j}|y^{\ell-1}, p_{\ell-1}^{\hat{s}\to i})\mathrm{P}(p_{\ell-1}^{\hat{s}\to i}|y^{\ell-1}).
\end{aligned}$$

The decomposition in (2.14) uses Bayes' rule and follows [1]. Fig. 2.4 identifies an example of states i and j used to compute the probability of $b_7^{i\to j}$.

By the Markov property, $\mathrm{f}(y_\ell|y^{\ell-1}, p_{\ell-1}^{\hat{s}\to i}, b_\ell^{i\to j}) = \mathrm{f}(y_\ell|b_\ell^{i\to j})$, which is the conditional p.d.f., related to the familiar Viterbi algorithm branch metric. Similarly, the second term is $\mathrm{P}(b_\ell^{i\to j}|y^{\ell-1}, p_{\ell-1}^{\hat{s}\to i}) = \mathrm{P}(b_\ell^{i\to j}|p_{\ell-1}^{\hat{s}\to i})$. With these simplifications, (2.14) becomes

$$\mathrm{P}(p_\ell^{\hat{s}\to j}|y^\ell) = \frac{\mathrm{f}(y_\ell|b_\ell^{i\to j})\mathrm{P}(b_\ell^{i\to j}|p_{\ell-1}^{\hat{s}\to i})\mathrm{P}(p_{\ell-1}^{\hat{s}\to i}|y^{\ell-1})}{\mathrm{f}(y_\ell|y^{\ell-1})}. \tag{2.15}$$

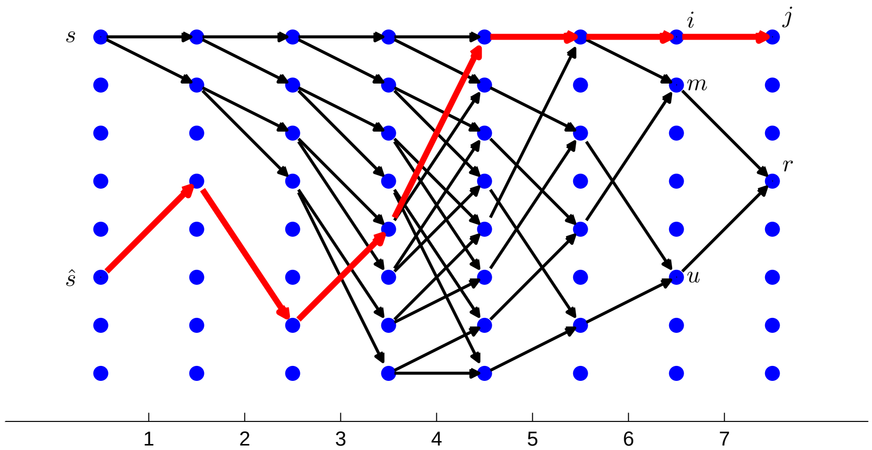


Figure 2.4: An example of a trellis for a rate-$1/n$ code with $\nu = 3$ memory elements is shown for $\ell = 1, 2, \ldots, 7$. The red branches show the candidate path from its beginning at state $\hat{s}$ to its arrival at state j. The black branches show all of the paths originating at state s from their beginning to their arrival at state r. Note that the figure does not show the final stage of the tail-biting trellis where all paths must return to their starting states.

The denominator can be expressed as a sum over all branches in the trellis $\mathcal{T}_\ell$ at time ℓ, where each branch from m to r is denoted by a pair $(m, r) \in \mathcal{T}_\ell$:

$$\mathrm{f}(y_\ell | y^{\ell-1}) = \sum_{(m,r)\in\mathcal{T}_\ell} \mathrm{f}(y_\ell | y^{\ell-1}, b_\ell^{m\to r}) \mathrm{P}(b_\ell^{m\to r} | y^{\ell-1}) \tag{2.16}$$

$$= \sum_{(m,r)\in\mathcal{T}_\ell} \mathrm{f}(y_\ell | b_\ell^{m\to r}) \mathrm{P}(b_\ell^{m\to r} | y^{\ell-1}). \tag{2.17}$$

The derivation thus far has followed [1], which focused on terminated convolutional codes.

For tail-biting codes, we can further expand the term $\mathrm{P}(b_\ell^{m\to r} | y^{\ell-1})$ by summing over all the possible starting states s' as follows:

$$\mathrm{P}(b_\ell^{m\to r} | y^{\ell-1}) = \sum_{s'} \mathrm{P}(b_\ell^{m\to r}, \bar{p}_{\ell-1}^{s'\to m} | y^{\ell-1}) \tag{2.18}$$

$$= \sum_{s'} \mathrm{P}(b_\ell^{m\to r} | y^{\ell-1}, \bar{p}_{\ell-1}^{s'\to m}) \mathrm{P}(\bar{p}_{\ell-1}^{s'\to m} | y^{\ell-1})$$

$$= \sum_{s'} \mathrm{P}(b_\ell^{m\to r} | \bar{p}_{\ell-1}^{s'\to m}) \mathrm{P}(\bar{p}_{\ell-1}^{s'\to m} | y^{\ell-1}), \tag{2.19}$$

where the last equality follows from the Markov property $\mathrm{P}(b_\ell^{m\to r}|y^{\ell-1}, \bar{p}_{\ell-1}^{s'\to m}) = \mathrm{P}(b_\ell^{m\to r}|\bar{p}_{\ell-1}^{s'\to m})$. Thus, (2.17) becomes

$$\mathrm{f}(y_\ell|y^{\ell-1}) = \sum_{(m,r)\in\mathcal{T}_\ell} \mathrm{f}(y_\ell|b_\ell^{m\to r}) \times \sum_{s'} \mathrm{P}(b_\ell^{m\to r}|\bar{p}_{\ell-1}^{s'\to m})\mathrm{P}(\bar{p}_{\ell-1}^{s'\to m}|y^{\ell-1}). \tag{2.20}$$

The term $\mathrm{P}(b_\ell^{m\to r}|\bar{p}_{\ell-1}^{s'\to m})$ is the probability that the branch from state m to state r is correct, given that one of the paths that started at state s' and arrived at state m at time $\ell-1$ is correct. $\mathrm{P}(b_\ell^{m\to r}|\bar{p}_{\ell-1}^{s'\to m}) = q^{-k}$ for $1 \le \ell \le K$ (i.e., all ℓ except for the last ν trellis segments in a convolutional code with $k = 1$). This is because there are q^k equiprobable next states for these values of ℓ.

Using the notation $r \Rightarrow s'$ to indicate there is a valid path from state r at time ℓ to state s' at time L, we define the following indicator function $\mathrm{I}(b_\ell^{m\to r}, s')$, which indicates that the trellis branch from state m to state r at trellis stage ℓ is a branch in a possible trellis path that terminates at s':

$$\mathrm{I}(b_\ell^{m\to r}, s') = \begin{cases} 1, & 1 \le \ell \le K, \quad (m,r) \in \mathcal{T}_\ell, \\ 1, & K+1 \le \ell \le L, \quad (m,r) \in \mathcal{T}_\ell, \quad r \Rightarrow s' \\ 0, & K+1 \le \ell \le L, \quad (m,r) \in \mathcal{T}_\ell, \quad r \not\Rightarrow s' \\ 0, & (m,r) \notin \mathcal{T}_\ell, \end{cases} \tag{2.21}$$

The branch-correct probabilities can now be written as

$$\mathrm{P}(b_\ell^{m\to r}|\bar{p}_{\ell-1}^{s'\to m}) = \begin{cases} \mathrm{I}(b_\ell^{m\to r}, s')\ q^{-k}, & 1 \le \ell \le K \\ \mathrm{I}(b_\ell^{m\to r}, s'), & K+1 \le \ell \le L \end{cases} \tag{2.22}$$

$$\mathrm{P}(b_\ell^{m\to r}|p_{\ell-1}^{\hat{s}\to m}) = \begin{cases} \mathrm{I}(b_\ell^{m\to r}, \hat{s})\ q^{-k}, & 1 \le \ell \le K \\ \mathrm{I}(b_\ell^{m\to r}, \hat{s}), & K+1 \le \ell \le L\ . \end{cases} \tag{2.23}$$

We now define the following normalization term for the ℓth trellis segment using the above indicators:

$$\Delta_\ell = \sum_{(m,r)\in\mathcal{T}_\ell} \mathrm{f}(y_\ell|b_\ell^{m\to r}) \sum_{s'} \mathrm{I}(b_\ell^{m\to r}, s')\mathrm{P}(\bar{p}_{\ell-1}^{s'\to m}|y^{\ell-1}) \tag{2.24}$$

$$= \begin{cases} \mathrm{f}(y_\ell|y^{\ell-1})\ q^k, & 1 \le \ell \le K \\ \mathrm{f}(y_\ell|y^{\ell-1}), & K+1 \le \ell \le L\ . \end{cases} \tag{2.25}$$

The Δ_ℓ normalization term includes most of (2.20) but excludes the potential q^{-k} in $\mathrm{P}(b_\ell^{m\to r}|\bar{p}_{\ell-1}^{s'\to m})$ because it cancels with $\mathrm{P}(b_\ell^{i\to j}|p_{\ell-1}^{\hat{s}\to i})$ in the numerator of (2.15). (Either both have q^{-k} or both are 1, depending only on ℓ.) Substituting (2.25) into (2.15), we have

$$\mathrm{P}(p_\ell^{\hat{s}\to j}|y^\ell) = \frac{1}{\Delta_\ell}\mathrm{f}(y_\ell|b_\ell^{i\to j})\mathrm{P}(p_{\ell-1}^{\hat{s}\to i}|y^{\ell-1}). \tag{2.26}$$

Thus, for the ℓth trellis segment, (2.26) expresses the candidate path-correct probability in terms of the candidate path-correct probability in the previous segment.

The corresponding expression for the overall path probabilities $\mathrm{P}(\bar{p}_\ell^{s\to r}|y^\ell)$ involves more terms. Instead of tracing a single candidate path through the trellis, we must add the probabilities of all the valid tail-biting paths incident on state r in segment ℓ as follows:

$$\mathrm{P}(\bar{p}_\ell^{s\to r}|y^\ell) = \frac{1}{\Delta_\ell} \sum_{m:(m,r)\in\mathcal{T}_\ell} \mathrm{f}(y_\ell|b_\ell^{m\to r}) \times \mathrm{I}(b_\ell^{m\to r}, s)\mathrm{P}(\bar{p}_{\ell-1}^{s\to m}|y^{\ell-1}). \tag{2.27}$$

The summation above is over the q^k incoming branches to state r. In the special case of a rate-$\frac{1}{n}$ binary code (q=2), there are 2 incoming branches, which we will label as (m,r) and (u,r), so (2.27) becomes

$$\begin{aligned}\mathrm{P}(\bar{p}_\ell^{s\to r}|y^\ell) =& \frac{1}{\Delta_\ell}\big[\mathrm{f}(y_\ell|b_\ell^{m\to r})\mathrm{I}(b_\ell^{m\to r}, s)\mathrm{P}(\bar{p}_{\ell-1}^{s\to m}|y^{\ell-1}) \\ &+ \mathrm{f}(y_\ell|b_\ell^{u\to r})\mathrm{I}(b_\ell^{u\to r}, s)\mathrm{P}(\bar{p}_{\ell-1}^{s\to u}|y^{\ell-1})\big].\end{aligned} \tag{2.28}$$

Fig. 2.4 illustrates how the paths from starting state s merge into state r at trellis segment $\ell = 7$.

2.4.2 PRC Algorithm Summary

The path probabilities are initialized as follows:

- $\mathrm{P}(p_0^{\hat{s}\to j}|y^0) = \mathrm{P}(\hat{s}) = q^{-\nu}$ if $\hat{s} = j$, or 0 otherwise.
- $\mathrm{P}(\bar{p}_0^{s\to r}|y^0) = \mathrm{P}(s) = q^{-\nu}$ if $s = r$, or 0 otherwise.

In each trellis-segment ℓ ($1 \leq \ell \leq L$), do the following:

1. For each branch $(m, r) \in \mathcal{T}_\ell$, compute the conditional p.d.f. $\mathrm{f}(y_\ell | b_\ell^{m\to r})$.
2. For each branch $(m, r) \in \mathcal{T}_\ell$ and each starting state s, compute the branch-valid indicator $\mathrm{I}(b_\ell^{m\to r}, s)$, as in (2.21).
3. Using the above values, compute the normalization constant Δ_ℓ, as in (2.24).
4. For the current state j of the candidate path, compute the candidate path-correct probability $\mathrm{P}(p_\ell^{\hat{s}\to j}|y^\ell)$, as in (2.26).
5. For each starting state s and each state r, compute the overall path-correct probabilities $\mathrm{P}(\bar{p}_\ell^{s\to r}|y^\ell)$, as in (2.27).

After processing all L stages of the trellis, the following meaningful quantities emerge:

- The posterior probability that the tail-biting candidate path from $\hat{s}$ to $\hat{s}$ is correct is $\mathrm{P}(p_L^{\hat{s}\to\hat{s}}|y^L) = \mathrm{P}(\hat{x}^L|y^L)$, which is the probability that the decoded word is correct, given the received sequence.
- The posterior word-error probability is then $1 - \mathrm{P}(p_L^{\hat{s}\to\hat{s}}|y^L) = 1 - \mathrm{P}(\hat{x}^L|y^L)$.
- The posterior probability that any of the tail-biting paths (any of the codewords) from s to s is correct is $\mathrm{P}(\bar{p}_L^{s\to s}|y^L) = \mathrm{P}(s|y^L)$, which is the state reliability desired for (2.5).

Numerical results of PRC are shown in Fig. 2.7 in Sec. 2.7.

2.5 The Tail-biting State-estimation Algorithm

The Post-decoding Reliability Computation described above relies on a separate decoder to identify the candidate path. If, on the other hand, we would like to compute the word-error probability of a tail-biting code without first having determined a candidate path and starting state, we may use the following Tail-Biting State-Estimation Algorithm (TB SEA). TB SEA computes the MAP starting state $\hat{s} = \arg\max_{s'} \mathrm{P}(s'|y^L)$, along with its reliability $\mathrm{P}(\hat{s}|y^L)$. ROVA($\hat{s}$) can then be used to determine the MAP codeword $\hat{x}_{\hat{s}}$ corresponding to starting state $\hat{s}$, as illustrated in Fig. 2.5.

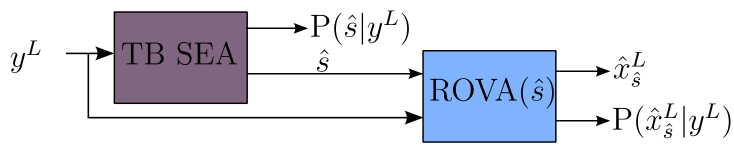


Figure 2.5: Block diagram of the Tail-Biting State-Estimation Algorithm (TB SEA), followed by ROVA($\hat{s}$) for the ML starting state $\hat{s}$.

PRC relied on tracing a single candidate path through the trellis and computing the candidate path-correct probability, as in (2.26). However, the overall path-correct probabilities in (2.27) do not rely on the candidate path or its probability. The proposed TB SEA aggregates all the previous-segment path-correct probabilities $\mathrm{P}(\bar{p}^{s \to m}_{\ell-1}|y^{\ell-1})$ as in (2.27), without regard to a candidate path. As a result, TB SEA replaces the traditional add-compare-select operations of the Viterbi algorithm with the addition of all the path probabilities merging into a state that emanate from a particular origin state. Once the entire trellis has been processed, the state reliabilities are compared and the MAP starting state is selected.

2.5.1 TB SEA Algorithm Summary

The path probabilities are initialized as follows:

- $\mathrm{P}(\bar{p}^{s \to r}_0|y^0) = \mathrm{P}(s) = q^{-\nu}$ if $s = r$, or 0 otherwise.

In each trellis-segment ℓ $(1 \leq \ell \leq L)$, do the following:

1. For each branch $(m, r) \in \mathcal{T}_\ell$, compute the conditional p.d.f. $\mathrm{f}(y_\ell | b_\ell^{m \to r})$.

2. For each branch $(m, r) \in \mathcal{T}_\ell$ and each starting state s, compute the branch-valid indicator $\mathrm{I}(b_\ell^{m \to r}, s)$, as in (2.21).

3. Using the above values, compute the normalization constant Δ_ℓ, as in (2.24).

4. For each starting state s and each state r, compute the overall path-correct probabilities $\mathrm{P}(\bar{p}_\ell^{s \to r} | y^\ell)$, as in (2.27).

After processing all L stages of the trellis, the following meaningful quantity emerges:

- The posterior probability that any of the tail-biting paths (any of the codewords) from s to s is correct is $\mathrm{P}(\bar{p}_L^{s \to s} | y^L) = \mathrm{P}(s | y^L)$.

TB SEA selects the starting state with the maximum value of $\mathrm{P}(s|y^L)$ (the MAP choice of starting state), yielding $\hat{s}$ and its reliability $\mathrm{P}(\hat{s}|y^L)$. Thus, TB SEA has selected the MAP starting state without explicitly evaluating all possible codewords (i.e., paths through the trellis). This result is not limited to error control coding; it can be applied in any context to efficiently compute the MAP starting state of a tail-biting, finite-state Markov process.

2.5.2 TB SEA + ROVA($\hat{s}$)

After finding the MAP starting state $\hat{s}$ with TB SEA, ROVA($\hat{s}$) may be used to compute the MAP codeword $\hat{x}_{\hat{s}}^L$ and $\mathrm{P}(\hat{x}_{\hat{s}}^L | y^L, \hat{s})$. We have used the subscript $\hat{s}$ to indicate that $\hat{x}_{\hat{s}}^L$ is the MAP codeword for the terminated code starting and ending in $\hat{s}$. The overall reliability $\mathrm{P}(\hat{x}_{\hat{s}}^L | y^L)$ can then be computed as in (2.5), which we have replicated below to show how the TB SEA and ROVA($\hat{s}$) provide the needed factors:

$$\mathrm{P}(\hat{x}|y) = \underbrace{\mathrm{P}(\hat{x}|y,\hat{s})}_{\text{computed by ROVA}(\hat{s})} \times \underbrace{\mathrm{P}(\hat{s}|y).}_{\text{computed by TB SEA}} \tag{2.29}$$

2.5.3 MAP States vs. MAP Codewords

Is it possible that the maximum a posteriori codeword $\hat{x}$ corresponds to a starting state other than the MAP state $\hat{s}$? The following theorem proves that the answer is no, given a suitable probability of error.

Theorem 1. *The MAP codeword $\hat{x}$ for a tail-biting convolutional code begins and ends in the MAP state $\hat{s}$, as long as* $\mathrm{P}(\hat{x}|y) > \frac{1}{2}$.

Proof. Consider a codeword $\hat{x}$ with $\mathrm{P}(\hat{x}|y) > \frac{1}{2}$. By (2.5), $\mathrm{P}(\hat{x}|y) = \mathrm{P}(\hat{x}|y, s_{\hat{x}})\mathrm{P}(s_{\hat{x}}|y)$, where $s_{\hat{x}}$ is the starting state of $\hat{x}$. This implies that $\mathrm{P}(s_{\hat{x}}|y) > \frac{1}{2}$. The MAP state is $\arg\max_{s'} \mathrm{P}(s'|y)$, which must be $s_{\hat{x}}$, since all other states s' must have $\mathrm{P}(s'|y) < \frac{1}{2}$. □

Theorem 1 shows that the application of TB SEA followed by the Viterbi algorithm (or the ROVA) will always yield the MAP codeword $\hat{x}$ of the tail-biting code, not just the MAP codeword for the terminated code starting in $\hat{s}$ (as long as the probability of error is less than $\frac{1}{2}$). In most practical scenarios, the word-error probability $(1 - \mathrm{P}(\hat{x}|y))$, even if unknown exactly, is much less than $\frac{1}{2}$, so the theorem holds. As a result, in these cases TB SEA selects the same codeword $\hat{x}$ as would the TB ROVA of Sec. 2.2.2, and computes the same reliability $\mathrm{P}(\hat{x}|y)$.

2.5.4 The TB BCJR Algorithm for State Estimation

While several related papers such as [31] and [32] have proposed ways to estimate the starting state of a tail-biting decoder, none computes exactly the posterior probability of the starting state, $\mathrm{P}(s|y^L)$, as described for TB SEA. Upon a first inspection, the tail-biting BCJR (TB

BCJR) of [26] appears to provide a similar method of computing this probability. Applying the forward recursion of the BCJR algorithm provides posterior probabilities that are denoted as $\alpha_L(s) = \mathrm{P}(s|y^L)$ in [26]. Thus, it would appear that the state-estimation algorithm of Sec. 2.5.1 can be replaced by a portion of the TB BCJR algorithm. This would yield a significant decrease in computational complexity, from roughly $q^{2\nu}$ operations per trellis segment for TB SEA to q^{ν} for the TB BCJR. However, as will be shown in Sec. 2.7, the word-error performance of the tail-biting BCJR when used in this manner is significantly inferior to that of TB SEA.

As noted in [28] and [29], the tail-biting BCJR algorithm is an approximate symbol-by-symbol MAP decoder. It is approximate in the sense that the forward recursion of the TB BCJR in [26] does not strictly enforce the tail-biting restriction, allowing non-tail-biting "pseudocodewords" to appear and cause errors. [26] requires the probability distributions of the starting and ending states to be the same, which is a weaker condition than requiring all codewords to start and end in the same state. [28] and [29] have shown that when the tail-biting length L is large relative to the memory order, the suboptimality of the TB BCJR in terms of the bit-error rate is small. However, we are concerned with word-error performance in this chapter. We find that when the TB BCJR is used to estimate the initial state $\hat{s}$ and its probability $\mathrm{P}(\hat{s}|y^L)$, followed by ROVA($\hat{s}$) for the most likely state $\hat{s}$, the impact on word error is severe (Fig. 2.7). Frequent state-estimation errors prevent the Viterbi algorithm in the second phase from decoding to the correct codeword. Thus, the approximate tail-biting BCJR of [26] is not effective as a replacement for TB SEA when using the word-error criterion.

In contrast, the exact symbol-by-symbol MAP decoder for tail-biting codes in [30, Ch. 7] does enforce the tail-biting restriction, and has complexity on the same order as that of TB SEA. However, because the symbol-by-symbol MAP decoder selects the most probable input symbols while TB SEA + ROVA($\hat{s}$) selects the most probable input sequence, TB SEA + ROVA($\hat{s}$) is recommended for use in retransmission schemes that depend on the word-error

Table 2.1: Complexity per trellis segment of the proposed algorithms (disregarding branch metric computations).

Algorithm	Path metrics	Cand. prob.	Overall prob.	Additions	Multiplications	Divisions
Key modules of decoders						
VA(s)	q^ν	0	0	0	$q^{\nu+k}$	0
ROVA(s) [1]	q^ν	q^ν	q^ν	$2q^\nu(2q^k-1)-1$	$3q^{\nu+k}$	$2q^\nu$
PRC	0	1	$q^{2\nu}$	$q^{2\nu}(2q^k-1)-1$	$q^{2\nu+k}$	$q^{2\nu}+1$
TB SEA	0	0	$q^{2\nu}$	$q^{2\nu}(2q^k-1)-1$	$q^{2\nu+k}$	$q^{2\nu}$
Approx ROVA(s) [18]	q^ν	q^ν	0	$q^\nu(q^k-1)$	$q^{\nu+k}+1$	q^ν
Tail-biting decoders that provide reliability output						
TB ROVA	$q^{2\nu}$	$q^{2\nu}$	$q^{2\nu}$	$2q^{2\nu}(2q^k-1)-q^\nu$	$3q^{2\nu+k}$	$2q^{2\nu}$
TB SEA + ROVA($\hat{s}$)	q^ν	q^ν	$q^{2\nu}+q^\nu$	$(q^{2\nu}+2q^\nu)(2q^k-1)-2$	$q^{2\nu+k}+3q^{\nu+k}$	$q^{2\nu}+2q^\nu$
Approx TB ROVA	$q^{2\nu}$	$q^{2\nu}$	0	$q^{2\nu}(q^k-1)$	$q^{2\nu+k}+q^\nu$	$q^{2\nu}$

probability.

2.6 Complexity Analysis

Table 2.1 compares the complexity per trellis segment of each of the discussed algorithms, assuming that the conditional p.d.f. $\mathrm{f}(y_\ell|b_\ell^{m\to r})$ has already been computed for every branch in the trellis. The columns labeled ‘Path metrics’, ‘Cand. prob.’, and ‘Overall prob.’ refer to the number of quantities that must be computed and stored in every trellis segment, for the path metrics of the Viterbi algorithm, the candidate path probability of (2.26), and the overall path probability of (2.27), respectively. The number of operations per trellis segment required to compute these values is listed in the columns labeled ‘Additions’, ‘Multiplications’, and ‘Divisions’.

The ROVA(s) row of Table 2.1 corresponds to Raghavan and Baum’s ROVA [1] for a terminated code starting and ending in state s. The operations listed include the multiplications required for the path metric computations of the Viterbi algorithm for state s, VA(s).

The TB ROVA row represents performing the ROVA for each of the q^ν possible starting states as described in Sec. 2.2.2, so each of the quantities is multiplied by q^ν.

The PRC row corresponds to the proposed Post-decoding Reliability Computation of Sec. 2.4. The complexity incurred to determine the candidate path (e.g., by the BVA or the A* algorithm) is not included in this row and must also be accounted for, which is why no path metrics are listed for PRC. Compared to TB ROVA, due to combining computations into a single pass through the trellis, the complexity of PRC is reduced by approximately a factor of 2. This is because TB ROVA calculates a candidate path probability for each of the q^ν starting states (due to decoding to the ML codeword each time), whereas the combined trellis-processing of PRC involves only one candidate path. Both algorithms compute q^ν overall path probabilities, so the ratio of complexity is roughly $\frac{1+q^\nu}{q^\nu+q^\nu} \approx \frac{1}{2}$.

The reduction in complexity of TB SEA compared to PRC is modest, with slightly fewer multiplications and divisions required due to the absence of the candidate path calculations in TB SEA. Importantly, performing TB SEA followed by ROVA($\hat{s}$) for the ML state $\hat{s}$ is shown to be an improvement over TB ROVA for moderately large ν. TB SEA + ROVA($\hat{s}$) requires approximately one half the additions, one third the multiplications, and one half the divisions of TB ROVA. TB SEA's complexity reduction is partly due to the fact that it does not require the add-compare-select operations of the Viterbi algorithm, which TB ROVA performs for each starting state. Note also that the number of trellis segments processed in TB SEA is constant (L segments), whereas the number of trellis segments processed by many tail-biting decoders (e.g., the BVA) depends on the SNR.

Lastly, the computational costs of performing Fricke and Hoeher's simplified ROVA [18] are listed in the Approx ROVA(s) row, along with the tail-biting approximate version of Sec. 2.3 (Approx TB ROVA). In both of these cases, the word-error outputs are estimates. In contrast, TB ROVA and TB SEA + ROVA($\hat{s}$) compute the exact word-error probability of the received word.

For the special case of rate-$1/n$ binary convolutional codes with $\nu = 6$ memory elements,

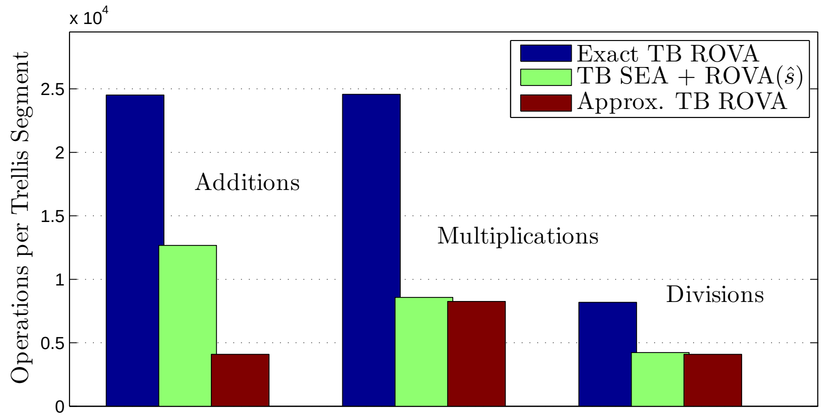


Figure 2.6: Examples of the computations per trellis segment for the tail-biting decoders listed in Table 2.1 corresponding to rate-$1/n$ binary convolutional codes, for $\nu = 6$ memory elements.

Table 2.2: Generator polynomials g_1, g_2, and g_3 corresponding to the simulated rate-1/3 tail-biting convolutional code. d_{free} is the free distance, $A_{d_{free}}$ is the number of nearest neighbors with weight d_{free}, and L_D is the analytic traceback depth.

ν	2^ν	g_1	g_2	g_3	d_{free}	$A_{d_{free}}$	L_D
6	64	117	127	155	15	3	21

Fig. 2.6 gives an example of the number of additions, multiplications, and divisions that must be performed per trellis segment for the three tail-biting decoders in Table 2.1. TB SEA + ROVA($\hat{s}$) is competitive with Approx TB ROVA in terms of the number of multiplications and divisions that must be performed, but Approx TB ROVA requires fewer additions than TB SEA + ROVA($\hat{s}$).

2.7 Numerical Results

2.7.1 Convolutional Code

Table 2.2 lists the rate-1/3, binary convolutional encoder polynomials from Lin and Costello [17, Table 12.1] used in the simulations. The number of memory elements is ν, 2^ν is the number of states, and $\{g_1, g_2, g_3\}$ are the generator polynomials in octal notation. The code selected has the optimum free distance d_{free}, which is listed along with the analytic traceback depth L_D [40]. $A_{d_{free}}$ is the number of nearest neighbors with weight d_{free}. The simulations in this section use a feedforward encoder realization of the generator polynomial.

2.7.2 Additive White Gaussian Noise (AWGN) Channel

For antipodal signaling (i.e., BPSK) over the Gaussian channel, the conditional density $\mathrm{f}(y_\ell | b_\ell^{i\to j})$ can be expressed as

$$\mathrm{f}(y_\ell | b_\ell^{i\to j}) = \prod_{m=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{[y_\ell(m) - x_\ell(m)]^2}{2\sigma^2} \right\}, \tag{2.30}$$

where $y_\ell(m)$ is the mth received BPSK symbol in trellis-segment ℓ, $x_\ell(m)$ is the mth output symbol of the encoder branch from state i to state j in trellis segment ℓ, and σ^2 is the noise variance. For a transmitter power constraint P, the encoder output symbols are $x_\ell(m) \in \{+\sqrt{P}, -\sqrt{P}\}$ and the energy per bit is $E_b = P\frac{n}{k}$. This yields an SNR equal to $P/\sigma^2 = 2\frac{k}{n}\frac{E_b}{N_0}$ when the noise variance is $\sigma^2 = N_0/2$.

2.7.3 Simulation Results

This section provides a comparison of the average word-error probability computed by the tail-biting reliability-output algorithms for the AWGN channel using the rate-1/3, 64-state tail-biting convolutional code listed in Table 2.2. The simulations in Fig. 2.7 use $L = 128$ input bits and 384 output bits. The 'Actual' curves in the figures show the fraction of codewords that are decoded incorrectly, whereas the 'Computed' curves show the word-error probability computed by the receiver. 'Actual' values are only plotted for simulations with more than 100 codewords in error.

Fig. 2.7 evaluates the performance of Sec. 2.2.2's TB ROVA and Sec. 2.4's PRC. In the figure, PRC is applied to the output of the Bidirectional Viterbi Algorithm (BVA), a suboptimal tail-biting decoder. The 'Actual' word-error performance of the suboptimal 'BVA' is slightly worse than that of the ML 'Exact TB ROVA', but the difference is not visible in Fig. 7. However, even though the bidirectional Viterbi decoder may choose a codeword other than the ML codeword, the posterior probability $\mathrm{P}(\hat{x}^L|y^L)$ computed by PRC is exact. Thus, PRC provides reliability information about the decoded word that the receiver can use as retransmission criteria in a hybrid ARQ setting.

Fig. 2.7 also shows the performance of the combined TB SEA + ROVA($\hat{s}$) approach in comparison with TB ROVA. As shown in Thm. 1, the word-error probability calculated by the computationally efficient TB SEA + ROVA(s) is identical to that of TB ROVA, except when the probability of error is extremely high (i.e., when $\mathrm{P}(\hat{x}|y) < \frac{1}{2}$). Even in the high-error regime, however, the difference is negligible.

The performance of the exact and approximate versions of TB ROVA is compared in Fig. 2.7. For each starting state s, the 'Exact TB ROVA' uses Raghavan and Baum's ROVA [1] and the 'Approx. TB ROVA' uses Fricke and Hoeher's simplified ROVA [18], as described in Sec. 2.3. The approximate approach results in an estimated word-error probability that is very close to the exact word-error probability. Both reliability computations invoke the same decoder, the tail-biting Viterbi algorithm ('TB VA'), so the 'Actual' curves are identical.

Finally, Fig. 2.7 also shows that when the forward recursion of the 'TB BCJR' of [26] is used to estimate the starting/ending state, there is a severe word-error penalty for disregarding the tail-biting restriction, as discussed in Sec. 2.5.4. The 'TB BCJR' simulations used

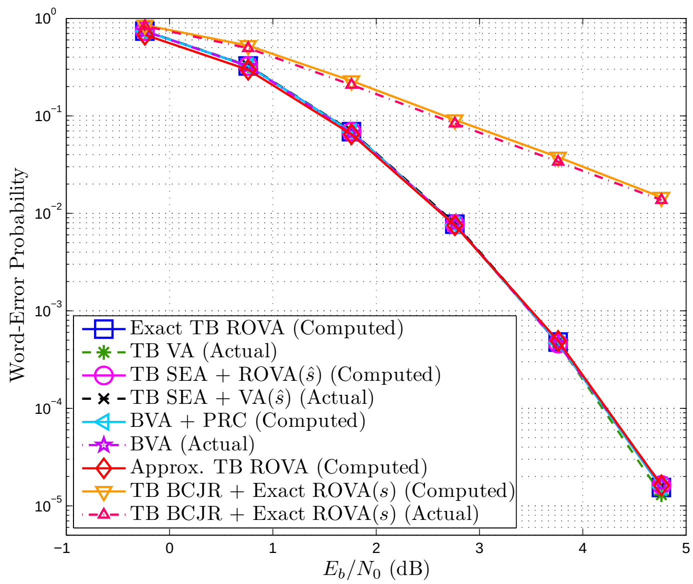


Figure 2.7: Computed and actual word-error probability of the exact TB ROVA, TB SEA followed by ROVA($\hat{s}$), and the Bidirectional Viterbi Algorithm (BVA) followed by the Post-decoding Reliability Computation (PRC). Also shown are the computed word-error probability estimates for Approx TB ROVA. All simulations use the rate-1/3, 64-state convolutional code listed in Table 2.2 with $L = 128$ input bits and 384 output bits and transmission over the AWGN channel. The 'Computed' values are the word-error probabilities calculated by the receiver (averaged over the simulation) and the 'Actual' values count the number of words decoded incorrectly. The 'TB BCJR' method of estimating the initial state is included for comparison, indicating that there is a severe penalty for disregarding the tail-biting restriction. Note that all curves except for the two 'TB BCJR' curves are almost indistinguishable.

one iteration through $L = 128$ trellis segments. Simulations with additional loops around the circular trellis did not improve the actual word-error probability, since the tail-biting condition was not enforced. Care should be taken when estimating the starting-state probability $\mathrm{P}(s|y^L)$ based on observations of y^L in multiple trellis-loops.

Fig. 2.8 provides a histogram of the word-error probabilities computed by the receiver for the rate-1/3, 64-state convolutional code listed in Table 2.2, with $L = 32$ input bits, 96 output bits and SNR 0 dB ($E_b/N_0 = 1.76$ dB). Fig. 2.8 illustrates that the exact and approximate TB ROVA approaches give very similar word-error probabilities, whereas the word-error probabilities computed by the tail-biting BCJR followed by ROVA($\hat{s}$) differ significantly. The difference in the histogram for the TB BCJR is due to poorer decoder performance. Frequent errors in the state-estimation portion of the tail-biting BCJR cause the word-error probability to be high.

2.8 Conclusion

We have extended the reliability-output Viterbi algorithm to accommodate tail-biting codes, providing several tail-biting reliability-output decoders. TB ROVA invokes Raghavan and Baum's ROVA for each possible starting state s, and then computes the posterior probability of the ML starting state, $\mathrm{P}(\hat{s}|y)$, in order to compute the overall word-error probability. We then demonstrated an approximate version of TB ROVA using Fricke and Hoeher's simplified ROVA. We introduced the Post-decoding Reliability Computation, which calculates the word-error probability of a decoded word, and the Tail-Biting State-Estimation Algorithm, which first computes the MAP starting state $\hat{s}$ and then decodes based on that starting state with ROVA($\hat{s}$).

A complexity analysis shows that TB SEA followed by ROVA($\hat{s}$) reduces the number of operations by approximately half compared to TB ROVA. Importantly, Theorem 1 proved that the word-error probability computed by TB SEA + ROVA($\hat{s}$) is the same as that

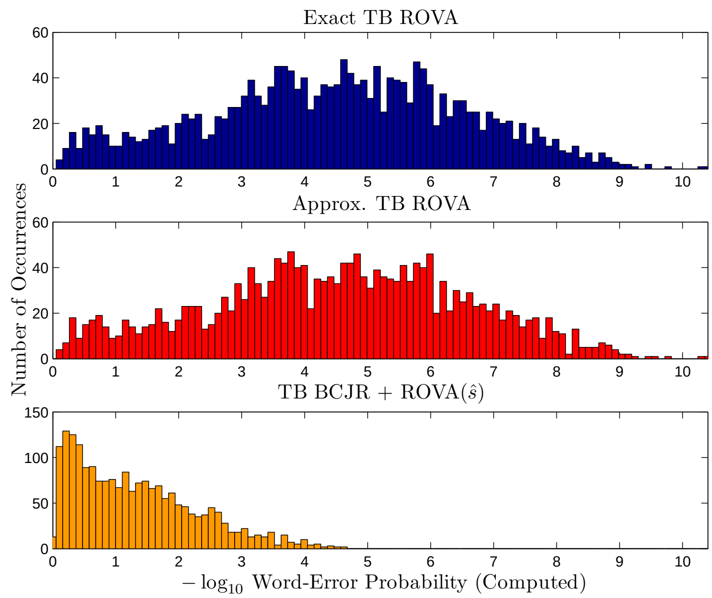


Figure 2.8: Histograms of the word-error probability, plotted on a logarithmic scale, computed by three reliability-output decoders: TB ROVA with ML decoding and exact reliability computations, Approx TB ROVA with ML decoding and approximate reliability computations, and the TB BCJR for sub-optimal estimation of the starting state $\hat{s}$ followed by ROVA($\hat{s}$). Each histogram includes simulations of the same 2000 transmitted codewords and noise realizations. The vertical axis is the number of times among the 2000 decoded words that the word-error probability falls within the histogram bin. The two ML decoders compute $\mathrm{P}(\hat{x}^L|y^L)$ for the same decoded word $\hat{x}^L$, whereas the suboptimal BCJR-based decoder decodes to a codeword that is not necessarily the same as $\hat{x}^L$. All simulations use the rate-1/3, 64-state convolutional code listed in Table 2.2, with $L = 32$ input bits, 96 output bits and SNR 0 dB ($E_b/N_0 = 1.76$ dB).

computed by TB ROVA in SNR ranges of practical interest. Because of this, TB SEA is a suitable tail-biting decoder to use in reliability-based retransmission schemes (i.e., hybrid ARQ), being an alternative to Approx TB ROVA.

CHAPTER 3

Low-latency Variable-length Coding with Feedback

3.1 Introduction

3.1.1 Overview and Related Literature

Despite Shannon's 1956 result [41] that noiseless feedback does not increase the (asymptotic) capacity of point-to-point, memoryless channels[1], feedback has other benefits for these channels that have made it a staple in modern communication systems. For example, feedback can simplify encoding and decoding operations and has been incorporated into incremental redundancy (IR) schemes proposed as early as 1974 [43]. Hagenauer's introduction of rate-compatible punctured convolutional (RCPC) codes allowed the same encoder to be used under varying channel conditions and used feedback to determine when to send additional coded bits [44]. Additionally, the information-theoretic benefit of feedback for reducing latency through a significant improvement in the error exponent, which governs the error probability as a function of blocklength, has been well understood for some time. See, for example, [45–48].

Perhaps the most important benefit of feedback is its ability to reduce the (average) blocklength required to approach capacity. Polyanskiy, Poor, and Verdú [24] quantified the backoff from capacity at short blocklengths without feedback, demonstrating that there is a severe penalty on the maximum achievable rate. Even when the best fixed-length block code

[1]Feedback does, however, provide bounded capacity-gain for the multiple access channel (MAC) and unbounded gain for the two-user Gaussian interference channel (IC) [42]. In this chapter, we focus on feedback in point-to-point channels.

is paired with an Automatic Repeat reQuest (ARQ) strategy, the maximum rate is slow to converge to the asymptotic (Shannon) capacity. However, when variable-length coding is used on channels with noiseless feedback, the maximum rate improves dramatically at short (average) blocklengths [25].

With variable-length coding, decoding after every individual received symbol is problematic from a practical perspective because it increases the decoder complexity tremendously. Instead of decoding once after all ℓ symbols have been received (as occurs with fixed-length codes), the receiver might attempt decoding after each symbol $1, 2, ..., \ell$, where ℓ is now a random variable.[2] Because the maximum ℓ is unbounded, the random coding in [25] assumes an infinite blocklength. Additionally, the round-trip propagation-delay incurred between channel uses in most cases will render such a scheme infeasible (since the transmitter waits for feedback of the previous channel output before sending the next coded symbol).

These practical considerations motivated Chen et al. [49, 50] to study the effects of periodic decoding (i.e., only decoding and sending feedback after every $I > 1$ symbols). Chen et al. concluded that the rate penalty incurred by limiting decoding times is a constant term. That is, throughput performance can approach capacity as long as the interval between periodic decoding times grows sub-linearly with the expected latency. Furthermore, the analysis in [49] and [50] used codebooks consisting of repeated finite-length-codes, showing that a code with maximum-length N can be used in a variable-length transmission scheme and provide the same performance as an infinite-length code (as in [25]), up to second order terms. The importance of these results is that "good" finite-length codes can still achieve rates approaching capacity at short blocklengths.

Contemporaneous with [24] and [25], Chen et al. [51, 52] showed that the relatively simple decoding of short-blocklength convolutional codes in an incremental redundancy setting

[2]In Polyanskiy's noiseless-termination paradigm [25], one could make the argument that the decoder only needs to attempt decoding once, when it receives the termination symbol. However, the transmitter would be decoding after each feedback symbol in order to determine when to terminate, so the system complexity still increases due to repeated decoding.

could match the throughput delivered by long-blocklength turbo codes. This led to the subsequent work in [49] and [50], which made connections between the theory and practical codes. Incremental redundancy and hybrid ARQ have been discussed elsewhere in the recent communication literature (e.g., [13, 15, 53]), but those papers are not focused on characterizing the maximum rate at short blocklengths. In [13], Visotsky et al. propose a hybrid ARQ algorithm that uses the reliability of received packets to determine the size of subsequent transmissions. However, in contrast to reliability-based retransmission protocols that compute the posterior probability of the received packet (e.g., those using the ROVA in [15, 23, 54]), [13] evaluates the reliability in terms of the average magnitude of the log likelihood-ratios of the received information bits. This approach requires a mapping between reliability metric and the word-error rate, which often requires approximations or bounding techniques. In contrast, the ROVA-based approach computes the word-error probability exactly.

3.1.2 Contributions

In this chapter, we investigate the performance of variable-length feedback (VLF) codes with blocklengths less than 300 symbols. In the VLF coding framework of Polyanskiy et al. [25], the receiver determines when to stop transmission and informs the transmitter via noiseless feedback. This is different from the variable-length feedback codes with termination (VLFT) introduced in [25] and explored in [49, 50], which use a special termination symbol on the forward channel to indicate when to stop. The transmitter's control of the stopping time enables zero-error VLFT codes, whereas VLF codes always have a nonzero (average) probability of error. The intent of the special termination-symbol is to model practical systems that have a highly-reliable control-channel that can effectively be considered noise-free, which allows the communication and termination aspects to be considered separately. From a simulation perspective, the termination symbol in the VLFT framework is equivalent to having a genie at the receiver that informs the decoder when it has received a sufficient number of noisy symbols to decode correctly.

However, the cost of knowing when the receiver has decoded successfully, either via termination symbols or common error-detection techniques such as CRCs, may be negligible for large blocklengths but is significant for small blocklengths [55]. In fact, ignoring this cost in the VLFT framework allows rates higher than the Shannon capacity when the average blocklength is small [25]. As a consequence, for the especially short blocklength regime (i.e., less than 100 symbols), the VLFT approach is not a realistic model.

In this dissertation, we evaluate the short-blocklength performance of two categories of VLF codes: 1) decision-feedback coding schemes, for which feedback is only used to inform the transmitter when to stop (also called stop-feedback), and 2) information-feedback schemes, which allow the transmitter to adapt its transmission based on information about the previously received symbols.

There is a large gap between the lower and upper bounds in [25] on achievable rate at short blocklengths for VLF codes. This chapter demonstrates explicit decision-feedback and information-feedback coding schemes that surpass the random-coding lower bound in [25]. These schemes use convolutional codes due to their excellent performance at short blocklengths. Numerical examples are given for the binary symmetric channel (BSC) and binary-input additive white Gaussian noise (BI-AWGN) channel.

The decision-feedback scheme in Williamson et al. [23] uses Raghavan and Baum's Reliability-Output Viterbi Algorithm (ROVA) [1] for terminated convolutional codes to compute the posterior probability of the decoded word and stops transmission when that word is sufficiently likely. This is similar to the reliability-based retransmission scheme in Fricke et al. [15], except that Fricke et al. is not focused on evaluating the short-blocklength performance. While the reliability-based retransmission scheme in [23] delivers high rates at relatively short blocklengths, the termination of the convolutional codes introduces rate loss at the shortest blocklengths.

Tail-biting convolutional codes, on the other hand, start and end in the same state, though that starting/ending state is unknown at the receiver. In exchange for increased decoding complexity, tail-biting codes do not suffer from rate loss at short blocklengths. However, Raghavan and Baum's ROVA [1] applies only to terminated convolutional codes. In [54], Williamson, Marshall and Wesel introduce a reliability-output decoder for tail-biting convolutional codes, called the tail-biting ROVA (TB ROVA). This chapter compares the reliability-based retransmission scheme using the TB ROVA with an alternative approach using Cyclic Redundancy Checks (CRCs), called code-based error detection. Both the ROVA and TB ROVA allow the decoder to request retransmissions without requiring parity bits to be sent for error detection.

When delay constraints or other practical considerations preclude decoding after every symbol, decoding after groups of transmitted symbols is required. Selecting the incremental transmission lengths that maximize the throughput is non-trivial, however. Sec. 3.6.7 provides a numerical optimization algorithm for selecting the m optimal blocklengths in a general m-transmission incremental redundancy scheme. Sec. 3.6.8 particularizes this algorithm to the reliability-based scheme using the TB ROVA.

Our information-feedback scheme, from Williamson et al. [56], borrows from the error-exponent literature and uses two-phase incremental redundancy to deliver high rates at short blocklengths. The communication phase uses tail-biting convolutional codes and the confirmation phase uses a fixed-length repetition code to confirm or deny the receiver's tentative decision. Sec. 3.6.9 particularizes the blocklength-selection algorithm of Sec. 3.6.7 to the two-phase scheme, using the assumption of rate-compatible sphere-packing error probabilities. Finally, we evaluate the short-blocklength performance of the active sequential hypothesis testing scheme in Naghshvar et al. [2, 3] and compare it to our two-phase incremental redundancy scheme. Whereas the two-phase scheme transmits packets, the hypothesis testing scheme uses feedback after every received symbol to adapt its subsequent transmissions.

The remainder of this chapter proceeds as follows: Sec. 3.1.3 introduces relevant notation. Sec. 3.2 provides a review of the fundamental limits for VLF codes from [25] and presents several extensions of the random-coding lower bound to VLF systems with "packets". Sec. 3.2 also includes improved VLF achievability bounds based on two-phase information feedback. Sec. 3.3 and Sec. 3.4 investigate decision-feedback and information-feedback schemes, respectively. Sec. 3.5 concludes the chapter.

3.1.3 Notation

In general, capital letters denote random variables and lowercase letters denote their realizations (e.g., random variable Y and value y). Superscripts denote vectors unless otherwise noted, as in $y^\ell = (y_1, y_2, \ldots, y_\ell)$, while subscripts denote a particular element of a vector: y_i is the ith element of y^ℓ. The expressions involving $\log(\ )$ and $\exp\{\ \}$ in information-theoretic derivations are valid for any base, but numerical examples use base 2 and present results in units of bits.

3.2 VLF Coding Framework

3.2.1 Finite-blocklength Information Theory

For finite-length block codes without feedback, Polyanskiy et al. [24] provide achievability (lower) and converse (upper) bounds on the maximum rate, along with a normal approximation of the information density that closely approximates both bounds for moderate blocklengths. In contrast to the tight characterization of the no-feedback case, there is a large gap between the lower and upper bounds for VLF codes at short average blocklengths presented in [25]. This section reviews the fundamental limits for VLF codes from [25] and provides extensions of the lower bound to repeat-after-N codes, which are similar in principle to the finite-length VLFT codes studied in [49, 50]. This framework will allow us to evaluate the short-blocklength performance of the decision-feedback schemes in Sec. 3.3 in terms of these fundamental limits. This section also provides extensions of the lower bound to information-feedback codes with two phases.

We assume there is a noiseless feedback channel. The practical implications of this assumption will be discussed further in Sec. 3.3. The noisy forward channel is memoryless, has input alphabet $\mathcal{X}$ and has output alphabet $\mathcal{Y}$. The channel satisfies

$$\mathrm{P}(Y_n|X^n, Y^{n-1}) = \mathrm{P}(Y_n|X_n) \tag{3.1}$$

$$= \mathrm{P}(Y_1|X_1) \ \forall \ n = 1, 2, \ldots \tag{3.2}$$

A discrete, memoryless channel (DMC) is a special case when $\mathcal{X}$ and $\mathcal{Y}$ are countable.

Definition 1. *(From [25]) An (ℓ, M, ϵ)* ***variable-length feedback (VLF)*** *code is defined by:*

- *A message $W \in \mathcal{W} = \{1, \ldots, M\}$, assumed to be equiprobable. (The positive integer M is the cardinality of the message set $\mathcal{W}$.)*
- *A random variable $U \in \mathcal{U}$ with $|\mathcal{U}| \leq 3$ and a probability distribution P_U on $\mathcal{U}$.[3] U represents common randomness that is revealed to both the transmitter and receiver before communication begins, which facilitates the use of random-coding arguments in the sequel.*
- *A sequence of encoder functions $f_n : \mathcal{U} \times \mathcal{W} \times \mathcal{Y}^{n-1} \to \mathcal{X}, n \geq 1$, which defines the nth channel input:*

$$X_n = f_n(U, W, Y^{n-1}). \tag{3.3}$$

- *A sequence of decoder functions $g_n : \mathcal{U} \times \mathcal{Y}^n \to \mathcal{W}, n \geq 1$, providing an estimate $\hat{W}_n$ of the message W at time n:*

$$\hat{W}_n = g_n(U, Y^n). \tag{3.4}$$

- *An integer-valued random variable $\tau \geq 0$, which is a stopping time of the filtration $\mathcal{G}_n = \sigma\{U, Y_1, \ldots, Y_n\}$. The stopping time satisfies*

$$\mathrm{E}[\tau] \leq \ell. \tag{3.5}$$

[3] *A proof of the upper bound on the cardinality of $\mathcal{U}$ is given in [25].*

- *A final decision computed at time τ, at which time the error probability must be less than ϵ $(0 \leq \epsilon \leq 1)$:*

$$\mathrm{P}[\hat{W}_\tau \neq W] \leq \epsilon. \tag{3.6}$$

Eq. (3.5) indicates that for an (ℓ, M, ϵ) VLF code, the expected length will be no more than ℓ. The rate R is given as $R = \frac{\log M}{\ell}$. In this chapter, we often refer to the average blocklength ℓ as the expected latency, which counts only the number of channel uses (i.e., transmitted symbols). The latency does not include delays inherent in practical systems, such as round-trip propagation delay, decoding delay, etc. A central assumption of this work is that there are no strict delay constraints, which would otherwise impose an upper limit on the maximum number of transmitted symbols before declaring an erasure. For an alternative treatment of feedback communication that analyzes the relationship between blocklength and delay, see work by Sahai (e.g., [57]).

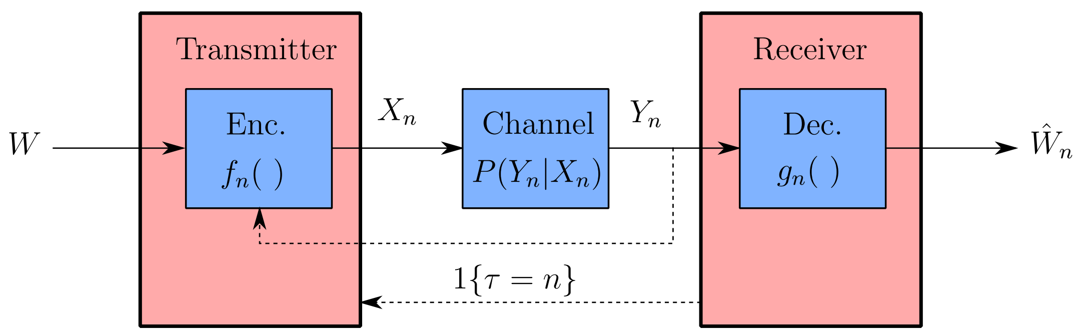


Figure 3.1: Illustration of the VLF coding framework, with message W, encoder (Enc.) output X_n at time n, memoryless channel output Y_n, and decoder (Dec.) estimate $\hat{W}_n$. The feedback link (dashed line) to the encoder is assumed to be noiseless. With decision feedback, the transmitter disregards the received symbols Y_n and only uses feedback to determine whether to stop (i.e., when $\tau = n$).

In the definition above, the receiver attempts to decode after each received symbol. Because the final decision is not computed until time $n = \tau$, the estimates $\hat{W}_n$ for $n < \tau$ can be thought of as tentative estimates that may not affect the decoding outcome. This differs slightly from [25], which does not require the decoder to compute the tentative estimates $g_n(U, Y^n)$ for $n < \tau$. We include the definition of $\hat{W}_n$ here for consistency with the reliability-based stopping approach in Sec. 3.3. The receiver uses some stopping rule (to be specified in Sec. 3.3) to determine when to stop decoding and informs the transmitter via feedback. This VLF coding framework is illustrated in Fig. 3.1.

Polyanskiy et al. [25] define a class of VLF codes called **stop-feedback codes** that satisfy:

$$f_n(U, W, Y^{n-1}) = f_n(U, W). \tag{3.7}$$

For stop-feedback codes, the encoded symbols are independent of the previously received noisy-channel outputs. The feedback link is used only to inform the transmitter when the receiver has determined that the transmission should terminate. This paradigm was referred to as **decision feedback** in early papers by Chang [58] and by Forney [22], the term which we will use in the sequel. Massey [59] referred to this case as a channel that is used without feedback.

Decision-feedback codes are practically relevant because they require only one bit of feedback for the receiver to communicate its termination decision or to request additional symbols. In the communication and networking literature, the feedback bit is usually referred to as an ACK (acknowledgment) or NACK (negative acknowledgment). A consequence of the decision-feedback restriction is that the codeword corresponding to message W, $\{X_n(W)\}_{n=1}^{\infty}$, can be generated before any symbols are transmitted over the channel, as the codeword does not depend on the realizations of $\{Y_n\}_{n=1}^{\infty}$.

Whereas noiseless feedback is in general not a practically sound assumption, a single bit of ACK/NACK feedback can be made virtually noiseless by use of a sufficiently strong error-correction code. For the feedback rate to be relatively low compared to the forward rate, however, (and therefore negligible in terms of the forward/feedback link budget) feedback must only be sent after decoding packets of multiple symbols, rather than after every received symbol. Packet-based VLF codes are explored further in the next subsection.

Another practical scenario that may motivate the assumption of noiseless feedback is an asymmetric communication link in which the transmitter's power constraint is significantly smaller than that of the receiver. Consider, for example, a battery-operated sensor node that, due to its limited transmit energy, imposes a low received SNR at the base station. If the base station's budget for decoding energy and feedback transmit energy are essentially unlimited, it can make sense to declare the feedback noiseless.

In contrast to decision-feedback codes, the class of VLF codes in which (3.7) is not generally true is referred to as **information-feedback** codes. The most general form of information feedback is for the receiver to communicate all of its received symbol values to the transmitter. With information feedback, the transmitter may take advantage of the feedback to adapt its transmissions to better refine the receiver's estimate, directing the receiver to the correct codeword. Naghshvar et al. [2, 3, 60–62] discuss this type of feedback in the context of active sequential hypothesis testing. In both [2] and [3], for example, Naghshvar et al. present a deterministic, sequential coding scheme for DMCs using information feedback that achieves the optimal error-exponent. In general, with information feedback the transmitter must wait for feedback from the receiver before generating each coded symbol.

Some examples of information feedback include [63], which provides feedback beyond simple ACKs and NACKs, [64], which feeds back the least reliable segments of a received convolutional code sequence, and [65], which feeds back the least reliable variable nodes of a non-binary LDPC code. The trellis-based decoding method described by Freudenberger and Stender in [64] enables the transmitter to resend only the unreliable segment(s) of a trellis, instead of retransmitting an entire block (as in type-I hybrid ARQ schemes) or retransmitting additional increments that are agnostic to the received sequence (as in type-II hybrid ARQ schemes with decision feedback). In both [63] and [64], the receiver does not send the received symbol Y_n as feedback, but sends some quantized representation of the received symbols.

In Vakilinia et al. [65], after decoding a high-rate non-binary LDPC code, the receiver uses feedback to inform the transmitter which symbols to resend. We classify these schemes as information feedback because unlike decision feedback, the transmitter uses the feedback to actively decide which coded symbol(s) to transmit next.

Ooi and Wornell [66, 67] describe a technique for feedback communication termed the compressed-error-cancellation framework, in which the receiver (noiselessly) sends back a block of received symbols Y^n and the transmitter compresses the original length-n block based on the feedback of Y^n. See Ooi [68] for additional references to early work on communication with feedback.

As described by Chang [58], in "information feedback systems, the receiver reports back in whole or in part the received information and the sender will decide whether or not he is satisfied with the information as received, and in the latter event, he will send corrective information". In contrast, Chang [58] describes the receiver's possible feedback messages in decision-feedback systems as either "please proceed" (ACK) or "please repeat" (NACK).

3.2.2 Fundamental Limits for VLF Codes

The following VLF achievability theorems make use of the information density $i(x^n; y^n)$ at blocklength n, defined as

$$i(x^n; y^n) = \log \frac{d\mathrm{P}(Y^n = y^n | X^n = x^n)}{d\mathrm{P}(Y^n = y^n)}. \tag{3.8}$$

We now restate Polyanskiy et al.'s achievability result for VLF codes:

Theorem 2 (Random-coding lower bound [25, Theorem 3])**.** *For a scalar $\gamma > 0$, $\exists$ an (ℓ, M, ϵ) VLF code satisfying*

$$\ell \leq \mathrm{E}[\tau], \tag{3.9}$$

$$\epsilon \leq (M-1)\mathrm{P}[\bar{\tau} \leq \tau], \tag{3.10}$$

where γ is a threshold for the hitting times τ and $\bar{\tau}$:

$$\tau = \inf\{n \geq 0 : i(X^n; Y^n) \geq \gamma\} \tag{3.11}$$

$$\bar{\tau} = \inf\{n \geq 0 : i(\bar{X}^n; Y^n) \geq \gamma\}. \tag{3.12}$$

In (3.12), $\bar{X}^n$ is distributed identically to X^n according to $\mathrm{P}(X^n)$, but is independent of (X^n, Y^n).

Although the random-coding lower bound in Thm. 2 gives an upper bound on the average blocklength ℓ corresponding to the maximum achievable rate for codes with cardinality M and error probability ϵ, it is not always straightforward to compute the achievable $(\ell, \frac{\log M}{\ell})$ pairs. In [23, Appendix B], Williamson et al. provide a method for computing these blocklength-rate pairs, based on numerical evaluation of an infinite sum. For the AWGN channel, each term in the sum requires a 3-dimensional numerical integration. In Sec. 3.6.1 of this chapter, we describe a different method of computing the average stopping time $\mathrm{E}[\tau]$ in (3.9) based on Wald's equality [69, Ch. 5]. This technique is computationally simpler and does not suffer from numerical precision issues that arise in evaluating an infinite sum. As explained in Sec. 3.6.1, the new method applies only to channels with bounded information density $i(X_n; Y_n)$ (e.g, the BSC or BI-AWGN channel, but not the AWGN channel with real-valued inputs).

It is straightforward to extend Polyanskiy et al.'s random-coding lower bound [25] to VLF codes derived from repeating length-N mother codes, which we will show in Cor. 1. We begin by defining (ℓ, M, ϵ) repeat-after-N VLF codes, for which the coded symbols for $n > N$ repeat the first N symbols. Let $r \in \{1, \ldots, N\}$ be the index within each block of N symbols, i.e., $n = sN + r$ for some $s \in \{0, 1, \ldots\}$.

Definition 2. *An (ℓ, M, ϵ) repeat-after-N VLF code is defined as in Def. 1, except the following are different:*

- *A sequence of encoder functions* $f_r : \mathcal{U} \times \mathcal{W} \times \mathcal{Y}_{sN+1}^{n-1} \to \mathcal{X}$*, which defines the nth channel input, where* $r \in \{1, \dots, N\}$*,* $n \geq 1$*,* $s = \lfloor \frac{n-1}{N} \rfloor$*:*

$$X_n = f_r(U, W, Y_{sN+1}^{n-1}) \,. \tag{3.13}$$

- *A sequence of decoder functions* $g_r : \mathcal{U} \times \mathcal{Y}_{sN+1}^{n} \to \mathcal{W}$*, providing an estimate* $\hat{W}_n$ *of the message* W*, where* $r \in \{1, \dots, N\}$*,* $n \geq 1$*:*

$$\hat{W}_n = g_r(U, Y_{sN+1}^{n}). \tag{3.14}$$

A practical consequence of this definition is that for decision-feedback repeat-after-N codes, only N unique coded symbols need to be generated for each message, due to the following property:

$$X_n = f_r(U, W). \tag{3.15}$$

Because the decoder only uses the received symbols from the current length-N block, we define the following modified information density:

$$i_N(X^n; Y^n) = \log \frac{d\mathrm{P}(Y_{sN+1}^n = y_{sN+1}^n | X_{sN+1}^n = x_{sN+1}^n)}{d\mathrm{P}(Y_{sN+1}^n = y_{sN+1}^n)} \tag{3.16}$$

$$= \log \frac{d\mathrm{P}(Y^r = y_{sN+1}^n | X^r = x_{sN+1}^n)}{d\mathrm{P}(Y^r = y_{sN+1}^n)}. \tag{3.17}$$

Corollary 1 (Random-coding lower bound for repeat-after-N codes)**.** *Suppose that* N *is large enough such that* $\mathrm{P}[\tau \leq N] > 0$*. Then for a scalar* $\gamma > 0$*,* $\exists$ *an* (ℓ, M, ϵ) *repeat-after-*N *VLF code satisfying* (3.10) *and*

$$\ell \leq \mathrm{E}[\tau] = \frac{\sum_{n=0}^{N-1} \mathrm{P}[\tau > n]}{1 - \mathrm{P}[\tau > N]}, \tag{3.18}$$

where γ *is a threshold for the hitting times* τ *and* $\bar{\tau}$*:*

$$\tau = \inf\{n \geq 0 : i_N(X^n; Y^n) \geq \gamma\} \tag{3.19}$$

$$\bar{\tau} = \inf\{n \geq 0 : i_N(\bar{X}^n; Y^n) \geq \gamma\}. \tag{3.20}$$

We will show in Sec. 3.3 that repeat-after-N VLF codes constructed by puncturing convolutional codes can deliver throughput surpassing that of the random-coding lower bound of Thm. 2, even when the random-coding lower bound does not use the repeat-after-N restriction. Similar behavior was seen in Chen et al. [49, 50], which explores the effect of finite-length codewords on the achievable rates of VLFT codes. The proof of Cor. 1 is in Sec. 3.6.3.

Thm. 1 can also be extended to accommodate repeat-after-N codes that permit decoding only at m specified intervals (modulo N_m): $n \in \{N_1, N_2, \dots, N_m, N_m + N_1, \dots\}$. Similar to the repeat-after-N setting, the coded symbols for $n > N_m$ repeat the first N_m symbols. We define $I_i = N_i - N_{i-1}$ as the transmission length of the ith transmission ($i = 1, \dots, m$), where $N_0 = 0$ for notational convenience. This framework models practical systems, in which decoding is attempted after groups of symbols instead of after individual symbols. The following corollary provides the achievability result for random coding with "packets".

Corollary 2 (Random-coding lower bound for m-transmission repeat-after-N_m codes)**.** *Suppose that N_m is large enough such that $\mathrm{P}[\tau \leq N_m] > 0$. Then for a scalar $\gamma > 0$, $\exists$ an (ℓ, M, ϵ) m-transmission, repeat-after-N_m VLF code satisfying* (3.10) *and*

$$\ell \leq \mathrm{E}[\tau] = \frac{\sum_{i=0}^{m-1} I_i \mathrm{P}[\tau > N_i]}{1 - \mathrm{P}[\tau > N_m]}, \tag{3.21}$$

where γ is a threshold for the hitting times τ and $\bar{\tau}$:

$$\tau = \inf \; \{n \geq 0 : i_N(X^n; Y^n) \geq \gamma\} \cap \{N_1, N_2, \dots, N_m, N_m + N_1, \dots\}, \tag{3.22}$$

$$\bar{\tau} = \inf \; \{n \geq 0 : i_N(\bar{X}^n; Y^n) \geq \gamma\} \cap \{N_1, N_2, \dots, N_m, N_m + N_1, \dots\}. \tag{3.23}$$

The proof of Cor. 2 is omitted. It closely follows that of Cor. 1 and relies on the fact that decoding can only occur at the specified intervals, so the expected stopping time in (3.21) is a sum of probabilities, weighted by the ith transmission length I_i.

3.2.3 Two-phase Information-feedback Lower Bounds

Note Thm. 2, Cor. 1, and Cor. 2 all use decision feedback to prove the existence of codes with the specified rates. We now present two achievability theorems for information-feedback coding schemes, in which the transmitter uses prior received symbols to determine the subsequent encoder outputs. Both are two-phase (communication and confirmation) schemes, for which the communication phase uses random coding as in Polyanskiy et al.'s random-coding lower bound. The first theorem considers a fixed-length confirmation phase and the second theorem considers a variable-length confirmation phase.

In the variable-length communication phase (for both bounds), the transmitter sends coded symbols according to the random codebook, as in Thm. 2. Once the receiver has decided that the first n received symbols are sufficiently reliable (i.e., $i(X^n;Y^n) \geq \gamma$ for some threshold $\gamma > 0$), the confirmation phase begins and the transmitter sends a coded confirmation message (ACK/NACK) on the forward channel to indicate to the receiver whether its tentative estimate is correct. The receiver informs the transmitter via noiseless feedback whether it received an ACK or a NACK. When the receiver decodes an ACK, the transmission terminates. If the receiver decodes a NACK, the receiver discards the received symbols and the process begins again with the transmission of a new communication phase.

There are two possible errors in the confirmation phase. An ACK may be decoded by the receiver incorrectly as NACK with probability $p_{a\to n}$. Similarly, a NACK can be mistaken for an ACK with probability $p_{n\to a}$. Correct reception of the confirmation phase has probability $p_{a\to a} = (1 - p_{a\to n})$ for ACK and $p_{n\to n} = (1 - p_{n\to a})$ for NACK.

Let $\mathcal{N}$ denote that event that the receiver decodes a NACK in the confirmation phase (following a single communication phase). The probability of this event is given by

$$\mathrm{P}(\mathcal{N}) = \underbrace{\mathrm{P}\{\hat{W}_{\tau^*} \neq W\}}_{\text{receiver's estimate incorrect}} p_{n\to n} + \underbrace{\mathrm{P}\{\hat{W}_{\tau^*} = W\}}_{\text{receiver's estimate correct}} p_{a\to n}, \tag{3.24}$$

where τ^* is the stopping time when the decoder decides that the ML codeword is sufficiently likely, as defined in Sec. 3.6.3. Upper bounding the first term in (3.24) as

$$\mathrm{P}\{\hat{W}_{\tau^*} \neq W\} \leq (M-1)\mathrm{P}[\bar{\tau} \leq \tau] \tag{3.25}$$

as in the random-coding lower bound (Thm. 2), and upper bounding the second term in (3.24) by 1, we have

$$\mathrm{P}(\mathcal{N}) \leq (1 - p_{n\to a})(M-1)\mathrm{P}[\bar{\tau} \leq \tau] + p_{a\to n}. \tag{3.26}$$

The probability of error after a communication-and-confirmation pair is $\mathrm{P}\{\hat{W}_{\tau^*} \neq W\}\ p_{n\,\to\,a}$. The overall probability of (undetected) error ϵ is given by

$$\begin{aligned} \epsilon &= \frac{\mathrm{P}\{\hat{W}_{\tau^*} \neq W\} p_{n\to a}}{1 - \mathrm{P}(\mathcal{N})} && (3.27)\\ &\leq \frac{(M-1)\mathrm{P}[\bar{\tau} \leq \tau] p_{n\to a}}{1 - \mathrm{P}(\mathcal{N})}. && (3.28) \end{aligned}$$

Note that the confirmation-phase crossover probabilities $p_{n\to a}$ and $p_{a\to n}$ may not be symmetric. A skewed hypothesis test may be employed by the receiver to bias against deciding NACK, since decoding NACKs incurs a latency cost due to starting over.

The first of our two achievability theorems for information-feedback codes uses a repetition code of length N_{conf} in the confirmation phase. Based on a variable-length random-coding communication phase and a fixed-length confirmation phase, we have the following theorem:

Theorem 3 (Two-phase information-feedback lower bound, fixed-length confirmation phase)**.** *For a scalar $\gamma > 0$ and integer $N_{conf} > 0$, there exists an (ℓ, M, ϵ) information-feedback VLF code satisfying*

$$\begin{aligned} \ell &\leq \frac{\mathrm{E}[\tau] + N_{conf}}{1 - \mathrm{P}(\mathcal{N})}, && (3.29)\\ \epsilon &\leq \frac{(M-1)\mathrm{P}[\bar{\tau} \leq \tau] p_{n\to a}}{1 - \mathrm{P}(\mathcal{N})}, && (3.30) \end{aligned}$$

where $\mathrm{P}(\mathcal{N})$ is the probability of the receiver decoding a NACK in the confirmation phase, which satisfies $\mathrm{P}(\mathcal{N}) < 1$, and γ is used as a threshold for determining the hitting times τ

and $\bar{\tau}$*:*

$$\tau = \inf\{n \geq 0 : i(X^n; Y^n) \geq \gamma\} \tag{3.31}$$

$$\bar{\tau} = \inf\{n \geq 0 : i(\bar{X}^n; Y^n) \geq \gamma\}. \tag{3.32}$$

While the stopping time τ is defined the same way in Thm. 3 (which uses information-feedback) and in the random-coding lower bound of Thm. 2 (decision-feedback), the confirmation phase in Thm. 3 permits a smaller value of the threshold γ to be used, which decreases the expected value $\mathrm{E}[\tau]$. When chosen properly, the additional latency of N_{conf} symbols in the confirmation block can decrease the overall two-phase latency. The proof of Thm. 3 is given in Sec. 3.6.4. Sec. 3.6.5 describes how to evaluate the bounds numerically.

Our second information-feedback scheme uses a binary, sequential hypothesis test in the second phase to decide between ACK and NACK at the decoder, instead of a fixed-length repetition code like in Thm. 3. A hypothesis testing rule known as Wald's sequential probability ratio test (SPRT) [70] provides tight bounds on the probability of error resulting from this sequential hypothesis test. The communication phase of this scheme is identical to the communication phase of Thm. 3. This scheme comes from Chen et al. [71], which analyzed both the error exponent and asymptotic expansion of the coding rate.

The two hypotheses for the confirmation phase are $\mathcal{H}_0$ (the receiver's tentative estimate $\hat{W}_\tau$ is correct and the transmitter has sent a forward ACK) and $\mathcal{H}_1$ (the receiver's estimate is incorrect and the transmitter has sent a forward NACK). $P_0(Y)$ and $P_1(Y)$ are the distributions of Y associated with hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively. The log likelihood-ratio of the ith observed channel output Y_i is defined as $\Lambda_i = \log \frac{P_1(Y_i)}{P_0(Y_i)}$, which yields a log likelihood-ratio Λ^n of the first n channel outputs Y^n given by

$$\Lambda^n = \log \frac{\prod_{i=1}^{n} P_1(Y_j)}{\prod_{i=1}^{n} P_0(Y_j)} \tag{3.33}$$

$$= \sum_{i=1}^{n} \Lambda_i. \tag{3.34}$$

In particular, under hypothesis $\mathcal{H}_0$, the transmitter sends $X = x_{\mathrm{c}}$ in each transmission, and sends $X = x_{\mathrm{e}}$ under hypothesis $\mathcal{H}_1$. (This is a variable-length repetition code.) The log likelihood-ratio is then $\Lambda_i = \log \frac{P(Y_i|X_i=x_{\mathrm{e}})}{P(Y_i|X_i=x_{\mathrm{c}})}$.

Let $t_0 < 0$ and $t_1 > 0$ be the decision thresholds for the receiver to decide $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively. (The thresholds will be specified later.) Transmission of the variable-length repetition code continues until one of these thresholds is crossed. Thus, the confirmation-phase stopping time μ is defined as

$$\mu = \inf\{n \geq 0 : \Lambda^n \leq t_0 \text{ or } \Lambda^n \geq t_1\}. \tag{3.35}$$

At time μ, the receiver accepts hypothesis $\mathcal{H}_0$ if $\Lambda^\mu \leq t_0$ or $\mathcal{H}_1$ if $\Lambda^\mu \geq t_1$. (This is the so-called SPRT.)

Wald's SPRT obeys the following bounds on the probability of error [70]:

$$P_0(\text{accept } \mathcal{H}_1) = p_{a\to n} \leq \exp\{-t_1\}, \tag{3.36}$$

$$P_1(\text{accept } \mathcal{H}_0) = p_{n\to a} \leq \exp\{t_0\}, \tag{3.37}$$

Note that the "error" corresponding to accepting hypothesis $\mathcal{H}_1$ denotes that the receiver decodes an ACK as a NACK and the two-phase scheme starts over, although no message errors occur. The "error" corresponding to accepting hypothesis $\mathcal{H}_0$ occurs when the receiver decodes a NACK as an ACK, which does result in an undetected error at the receiver. By (3.26), the probability of the receiver decoding a NACK is given by

$$\mathrm{P}(\mathcal{N}) \leq (M-1)\mathrm{P}[\bar{\tau} \leq \tau](1 - p_{n\to a}) + p_{a\to n} \tag{3.38}$$

$$\leq (M-1)\mathrm{P}[\bar{\tau} \leq \tau] + \exp\{-t_1\}. \tag{3.39}$$

Note that in (3.39), the upper bound from Wald's SPRT on $p_{n\to a}$ cannot be used to upper bound $\mathrm{P}(\mathcal{N})$, so $\mathrm{P}(\mathcal{N})$ only depends on the threshold t_1 (decide $\mathcal{H}_1$), not t_0 (decide $\mathcal{H}_0$).

Importantly, Wald's framework shows that any other sequential hypothesis test (i.e., besides the SPRT) with error probabilities $p'_{a\to n} \leq p_{a\to n}$ and $p'_{n\to a} \leq p_{n\to a}$ necessarily has expected length $\mathrm{E}[\mu'] \geq \mathrm{E}[\mu]$ [70]. Thus, given desired error probabilities in the confirmation phase, we can use Wald's SPRT to minimize the expected length $\mathrm{E}[\mu]$.

Putting this together with the random-coding lower bound for the communication phase, we have the following theorem:

Theorem 4 (Two-phase, information-feedback lower bound, variable-length confirmation phase)**.** *For scalars $\gamma > 0$, confirmation threshold $t_0 < 0$, and retransmit threshold $t_1 > 0$, there exists an (ℓ, M, ϵ) information-feedback VLF code satisfying*

$$\ell \leq \frac{\mathrm{E}[\tau] + \mathrm{E}[\mu]}{1 - \mathrm{P}(\mathcal{N})}, \tag{3.40}$$

$$\epsilon \leq \frac{(M-1)\mathrm{P}[\bar{\tau} \leq \tau] p_{n\to a}}{1 - \mathrm{P}(\mathcal{N})}, \tag{3.41}$$

$$\leq \frac{(M-1)\mathrm{P}[\bar{\tau} \leq \tau] \exp\{t_0\}}{1 - (M-1)\mathrm{P}[\bar{\tau} \leq \tau] - \exp\{-t_1\}}, \tag{3.42}$$

where μ is defined as in (3.35)*,* $\mathrm{P}(\mathcal{N})$ *is the probability of the receiver decoding a NACK in the confirmation phase, which satisfies* $\mathrm{P}(\mathcal{N}) < 1$*, and γ is used as a threshold for determining the hitting times τ and $\bar{\tau}$:*

$$\tau = \inf\{n \geq 0 : i(X^n; Y^n) \geq \gamma\} \tag{3.43}$$

$$\bar{\tau} = \inf\{n \geq 0 : i(\bar{X}^n; Y^n) \geq \gamma\}. \tag{3.44}$$

The proof of Thm. 4 is omitted. It closely follows that of Thm. 3, which is given in Sec. 3.6.4. Details for computing the expected stopping time of the confirmation phase, $\mathrm{E}[\mu]$, are provided in Sec. 3.6.6.

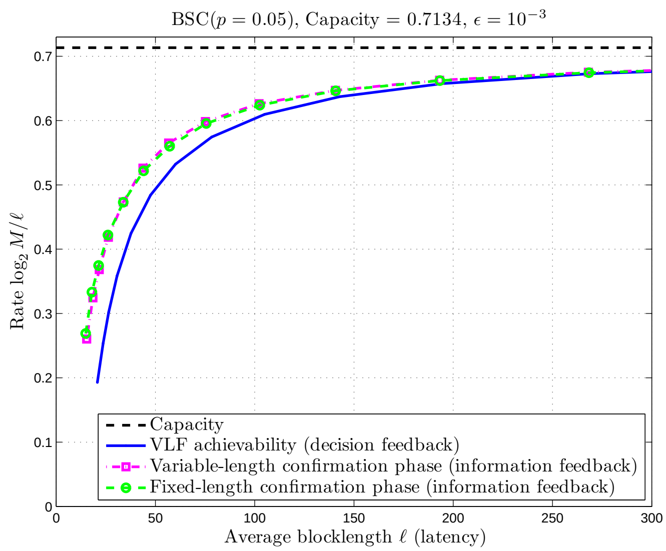


Figure 3.2: An example of the information-feedback lower bounds given in Thm. 3 (variable-length communication phase, fixed-length confirmation phase) and Thm. 4 (variable-length communication phase, variable-length confirmation phase) compared to the decision-feedback lower bound of Thm. 2 (variable-length communication phase only), for the BSC with crossover probability $p = 0.05$ and error probability $\epsilon = 10^{-3}$. Sec. 3.6.5 and Sec. 3.6.6 describe how to evaluate Thm. 3 and Thm. 4 numerically, respectively.

Fig. 3.2 demonstrates the improvement provided by the two-phase information-feedback lower bounds of Thms. 3 and 4 compared to the decision-feedback bound of Thm. 2. For this example of the BSC with crossover probability $p = 0.05$ and error constraint $\epsilon = 10^{-3}$, the confirmation phase (based on information feedback from the communication phase) improves the achievable rate for blocklengths less than approximately 150 bits, with the biggest improvement under 50 bits. In this region, random coding alone (with decision feedback) requires a substantial number of transmitted symbols to meet the ϵ error requirement, whereas the addition of a confirmation phase allows a smaller average value $\mathrm{E}[\tau]$ in the first phase. Past 150 bits, however, the two-phase information feedback and one-phase decision feedback bounds promise roughly the same rates. Interestingly, the throughput predicted by the fixed-length confirmation phase (Thm. 3) and the variable-length confirmation phase (Thm. 4) are not significantly different. This may be because of the upper bounds employed for $\mathrm{P}(\mathcal{N})$ in (3.39) and $\mathrm{E}[\mu]$ are not tight.

A separate example of the two-phase information-feedback bound given in Chen et al. [71] for a stricter error constraint of $\epsilon = 10^{-6}$ shows a more drastic improvement compared to random coding. This behavior agrees with the intuition that random coding and decision feedback can be beneficial for establishing a tentative estimate of the ML codeword, but that the active confirmation phase based on information feedback is key to meeting the ϵ error requirement.

Whereas the decision-feedback approach of Thm. 2 requires only one bit of feedback per forward-channel use, the information-feedback approach of Thms. 3 and 4 assume that the receiver feeds back each received symbol Y_n. For channels with non-binary outputs (e.g., the BI-AWGN channel), this would require a significant increase in the feedback rate. More importantly, in practical systems that use packet transmissions (as in Thm. 2), decision feedback requires the ACK/NACK bit to be sent only once per packet. Information feedback nonetheless assumes all received symbols Y^n are fed back. Alternatively, the receiver in the information-feedback setting may convey its tentative decoding estimate $\hat{W}_n$ (with cardinality M), but only once per communication phase.

The two-phase framework of Thms. 3 and 4 are considered information feedback because the transmitter uses the feedback from the first phase to select the confirmation symbols in the second phase. However, within each phase, the transmitter does not adapt its output based on the received symbols, so each phase can be categorized as using decision feedback only.

The active sequential hypothesis testing work of Naghshvar et al. [72] refers to the relative advantage of information-feedback coding versus decision-feedback coding as *adaptivity gain*, in reference to the benefit of the transmitter adapting to the realization of the channel. *Sequentiality gain* occurs when a variable-length coding strategy is used in place of fixed-length block coding.

3.2.4 Converse Bounds

Whereas the preceding theorems have all been lower bounds, we now review upper bounds on the maximum rate in the non-asymptotic regime. Polyanskiy et al.'s converse in [25] for VLF codes comes from [46, Lemmas 1 and 2]:

Theorem 5 ([25, Theorem 4]). *For an arbitrary DMC with capacity C and $0 \le \epsilon \le 1 - \frac{1}{M}$, any (ℓ, M, ϵ) VLF code satisfies:*

$$\log M \le \frac{\ell C + h_b(\epsilon)}{1-\epsilon}, \tag{3.45}$$

where $h_b(\epsilon)$ is the binary entropy function.

The concept of maximal relative entropy is central to Thm. 6 below. The maximal relative entropy C_1 is defined as

$$C_1 = \max_{x_1, x_2 \in \mathcal{X}} D\left(\mathrm{P}(Y|X = x_1) || \mathrm{P}(Y|X = x_2)\right). \tag{3.46}$$

When C_1 is finite, we have the following converse result, which is tighter than Thm. 5:

Theorem 6 ([25, Theorem 6]). *For a DMC with* $0<C\leq C_1<\infty$ *and* $0 < \epsilon \leq 1 - \frac{1}{M}$, *any* (ℓ, M, ϵ) *VLF code satisfies:*

$$\ell \geq \sup_{0<\xi\leq 1-\frac{1}{M}} \left[\frac{1}{C} \left(\log M - F_M(\xi) - \min\left(F_M(\epsilon), \frac{\epsilon}{\xi}\log M \right) \right) \right.$$
$$\left. + \left| \frac{1-\epsilon}{C_1} \log \frac{\lambda_1 \xi}{\epsilon(1-\xi)} - \frac{h_b(\epsilon)}{C_1} \right|^+ \right], \tag{3.47}$$

$$F_M(x) = x\log(M-1) + h_b(x), 0 \leq x \leq 1, \tag{3.48}$$

$$\lambda_1 = \min_{y,x_1,x_2} \frac{\mathrm{P}(Y=y|X=x_1)}{\mathrm{P}(Y=y|X=x_2)} \in (0,1). \tag{3.49}$$

As shown in Fig. 3.3(a) for the binary symmetric channel (BSC) with crossover probability $p = 0.05$, there is a considerable gap between the lower and upper bounds on the maximum rate at short blocklengths. This is in contrast to the finite-blocklength bounds for fixed-length codes without feedback, which are tight and can be approximated at moderate blocklengths ($e.g., n > 100$) by the normal approximation [24]. Because of the gap in the fundamental limits for VLF codes, it is not clear what finite-blocklength performance is achievable. To investigate this, we present two deterministic coding schemes in Sec. 3.3 and Sec. 3.4 and analyze their performance in the short-blocklength regime ($n < 300$). In both schemes, we fix the target error probability ϵ and explore what rate can be achieved at short blocklengths.

3.3 Decision-feedback Codes

3.3.1 Reliability-based Error Detection

This section presents a decision-feedback coding scheme using reliability-based error detection. Recall that in any decision-feedback scheme, the receiver must have a stopping rule that guarantees the error probability is less than ϵ. Many practical systems use CRCs for explicit error detection, sometimes referred to as a code-based approach. However, at short blocklengths, the latency overhead of a CRC strong enough to meet the ϵ requirement may result in a significant rate penalty. Here we investigate reliability-based error detection, choosing to stop decoding when the posterior probability of the decoded word is at least $(1-\epsilon)$. This does not require additional coded symbols to be sent for error detection. We show at the end of this section that a reliability-based decoding scheme can outperform CRC-based error detection at short blocklengths and can deliver throughput surpassing the random-coding lower bound.

For practical purposes, we consider only repeat-after-N codes in this section. After receiving the nth transmitted symbol, the receiver computes the posterior probability (the reliability) of the maximum a posteriori (MAP) message $\hat{W}_n$, where

$$\hat{W}_n = \arg\max_{i \in \mathcal{W}} \mathrm{P}(W = i | Y_{sN+1}^n). \tag{3.50}$$

The stopping rule for the reliability-based (RB) retransmission scheme is defined according to:

$$\tau^{(\mathrm{RB})} = \inf\{n \geq 0 : \mathrm{P}(W = \hat{W}_n | Y_{sN+1}^n) \geq 1 - \epsilon\}. \tag{3.51}$$

In principle, the computation of the M posterior probabilities in (3.50) can be performed for any code, such as LDPC codes. However, even for moderate blocklengths, this may not be computationally feasible in general, similar to the complexity challenge of ML decoding. Fortunately, for terminated convolutional codes, Raghavan and Baum's Reliability-Output Viterbi Algorithm (ROVA) [1] gives an efficient method to compute $\mathrm{P}(W = \hat{W}_n | Y_{sN+1}^n)$, the posterior probability of the MAP message[4]. The computational complexity of the ROVA is linear in the blocklength and exponential in the constraint length, on the same order as that of the Viterbi Algorithm. This allows the receiver to implement the stopping rule in (3.51) without explicitly evaluating all M posterior probabilities. Due to this construction, the overall probability of error in the reliability-based stopping scheme will satisfy the ϵ

[4]When the source symbols are equiprobable, there is a one-to-one correspondence between the MAP message and the ML codeword, the latter of which is identified by both the Viterbi Algorithm and the ROVA.

constraint:

$$\mathrm{P}[\hat{W}_{\tau^{(\mathrm{RB})}} \neq W] = \mathrm{E}\big[1 - \mathrm{P}[\hat{W}_{\tau^{(\mathrm{RB})}} = W | Y_{sN+1}^{\tau^{(\mathrm{RB})}}]\big] \leq \epsilon. \tag{3.52}$$

An alternative algorithm for computing the MAP message probability for terminated convolutional codes is given by Hof et al. [21], which provides a modification to the Viterbi Algorithm that permits decoding with erasures according to Forney's generalized decoding rule [22]. The MAP message probability can also be computed approximately by Fricke and Hoeher's simplified (approximate) ROVA [18], which is an ML sequence decoder that computes an estimated posterior probability.

However, terminated convolutional codes suffer from rate loss at short blocklengths, as described earlier, and Raghavan and Baum's ROVA [1] does not permit decoding of throughput-efficient tail-biting convolutional codes (TBCCs). Williamson et al. [54]'s tail-biting ROVA describes how to compute the posterior probability of MAP messages corresponding to tail-biting codewords. In the simulations that follow, we use both the ROVA for terminated codes and, when computational complexity permits, the TB ROVA for tail-biting codes. In particular, we implement an efficient version of the TB ROVA called the Tail-Biting State-Estimation Algorithm (TB SEA) from [54] that reduces the number of computations but still computes the MAP message probability exactly[5].

The details of our decision-feedback scheme are as follows. Similar to Fricke and Hoeher [15]'s reliability-based retransmission criteria for hybrid ARQ, if the computed word-error probability at blocklength n is greater than the target ϵ, the decoder signals that additional coded symbols are required (sends a NACK), and the transmitter sends another coded symbol. When the word-error probability is less than ϵ, the decoder sends an ACK, and transmission stops. We encode a message with $k = \log M$ message symbols into a mother codeword of length N. One symbol is transmitted at a time, using pseudo-random, rate-compatible

[5]The TB SEA and TB ROVA compute the same probability as long as $\mathrm{P}(W = \hat{W}_n | Y) > \frac{1}{2}$. In the proposed reliability-based retransmission scheme with $\epsilon < \frac{1}{2}$, this condition is met for $\tau^{(\mathrm{RB})} = n$, so the TB SEA is an ML sequence decoder.

puncturing of the mother code. At each decoding opportunity, the receiver uses all received symbols to decode and computes the MAP message probability. If the receiver requests additional redundancy after N symbols have been sent, the transmitter begins resending the original sequence of N symbols and decoding starts from scratch. (This is a repeat-after-N VLF code.) While some benefit can be accrued by retaining the N already-transmitted symbols (for example, by Chase code combining), we do not exploit this opportunity in our scheme for the sake of simplicity.

Similar to the random-coding lower bound for repeat-after-N codes in Cor. 1, we can express the latency $\lambda^{(\mathrm{RB})}$ and the throughput $R_t^{(\mathrm{RB})}$ of the proposed scheme as

$$\lambda^{(\mathrm{RB})} \leq \frac{1 + \sum_{i=1}^{N-1} P_{\mathrm{NACK}}(i)}{1 - P_{\mathrm{NACK}}(N)}, \tag{3.53}$$

$$R_t^{(\mathrm{RB})} = \frac{k}{\lambda^{(\mathrm{RB})}}(1 - P_{\mathrm{UE}}), \tag{3.54}$$

where $P_{\mathrm{NACK}}(i)$ is the probability that a NACK is generated because the MAP message probability is less than $(1-\epsilon)$ when i coded symbols (modulo N) have been received, and P_{UE} is the overall probability of undetected error. Note $P_{\mathrm{UE}} \leq \epsilon$ by definition of the stopping rule, as shown in (3.52). We obtain $P_{\mathrm{NACK}}(i)$ and P_{UE} via simulation in the following section and plot the resulting $(\lambda^{(\mathrm{RB})}, R_t^{(\mathrm{RB})})$ pairs. We have included the factor $(1 - P_{\mathrm{UE}})$ in the throughput expression to emphasize that we are only counting the messages that are decoded successfully at the receiver (i.e., the goodput).

In the following subsection, we will compare simulations of the reliability-based retransmission scheme with the random-coding lower bound of Thm. 2. For the sake of clarity, we first provide a comparison of the stopping rules in each scheme. The receiver in Thm. 2 stops decoding when the information density crosses a threshold γ as in (3.11), which can be expressed as

$$\tau = \inf\left\{ n \geq 0 : \frac{d\mathrm{P}(X^n = x^n | Y^n = y^n)}{d\mathrm{P}(X^n = x^n)} \geq \exp\{\gamma\} \right\}. \tag{3.55}$$

With decision feedback and a random codebook, each encoder output $x_n \in \mathcal{X}$ is equally likely and the sequence probability can be written in terms of the product distribution: $d\mathrm{P}(X^n = x^n) = \prod_{i=1}^{n} d\mathrm{P}(X_i = x_i)$. (For the BSC, $d\mathrm{P}(X^n = x^n) = 2^{-n}$.)

In contrast, for a deterministic codebook (e.g., for a convolutional code), there is a one-to-one mapping between the message $W \in \mathcal{W}$ and its length-N codeword $X^N(W)$. For a deterministic codebook, the reliability-based stopping rule in (3.51) can be rewritten as follows:

$$\tau^{(\mathrm{RB})} = \inf\left\{ n \geq 0 : \frac{\mathrm{P}(W = \hat{W}_n | Y_{sN+1}^n)}{\mathrm{P}(W = \hat{W}_n)} \geq \frac{1-\epsilon}{\mathrm{P}(W = \hat{W}_n)} \right\} \tag{3.56}$$

$$= \inf\left\{ n \geq 0 : \log \frac{d\mathrm{P}(W = \hat{W}_n | Y_{sN+1}^n)}{d\mathrm{P}(W = \hat{W}_n)} \geq \log(1-\epsilon) + \log M \right\}, \tag{3.57}$$

where we have used the fact that $\mathrm{P}(W = \hat{W}_n) = \frac{1}{M}$ since the messages are equiprobable. We define a modified information density for a deterministic mother code with length N as

$$i(W; Y_{sN+1}^n) = \log \frac{d\mathrm{P}(W = \hat{W}_n | Y_{sN+1}^n)}{d\mathrm{P}(W = \hat{W}_n)}. \tag{3.58}$$

Putting this together with (3.57), the reliability-based stopping rule becomes

$$\tau^{(\mathrm{RB})} = \inf\{n \geq 0 : i(W; Y_{sN+1}^n) \geq \log(1-\epsilon) + \log M\}. \tag{3.59}$$

In this manner, the threshold for the modified information density in the reliability-based retransmission scheme is $\gamma^{(\mathrm{RB})} = \log(1-\epsilon) + \log M$. (Keep in mind that the information density defined in (3.58) relates the information between the received symbols and the message, rather than the received symbols and the transmitted symbols. The threshold $\gamma^{(\mathrm{RB})}$ is in terms of the information density in (3.58).)

3.3.2 Convolutional Code Polynomials

This section briefly lists the convolutional code polynomials used in the subsequent VLF coding simulations. For the decision-feedback coding scheme in Sec. 3.3, we use both terminated convolutional codes and tail-biting convolutional codes. The former terminate in

Table 3.1: Generator polynomials g_1, g_2, and g_3 corresponding to the rate 1/3 convolutional codes used in the VLF simulations. d_{free} is the free distance, $A_{d_{\text{free}}}$ is the number of codewords with weight d_{free}, and L_D is the analytic traceback depth.

# Memory Elements, ν	# States $s = 2^\nu$	Polynomial (g_1, g_2, g_3)	d_{free}	$A_{d_{\text{free}}}$	L_D
6	64	(117, 127, 155)	15	3	21
10	1024	(2325, 2731, 3747)	22	7	34

a known state (usually the all-zeros state) and the latter have a common starting/ending state, which is unknown at the receiver. For terminated convolutional codes with large constraint lengths ($\nu + 1$), a significant rate penalty (especially at short blocklengths) is incurred due to the ν symbols that must be transmitted in order to arrive at the termination state. For convolutionally coded sequences with k information bits and N coded symbols, the rate of the terminated code is $R = \frac{k+\nu}{N}$, meaning that the effective information-rate is $R_{\text{eff}} = \frac{k}{N}$ and the rate-loss factor is $\frac{\nu}{k+\nu}$. In contrast, there is no rate loss for tail-biting codes, but a higher decoder complexity is required to determine the starting/ending state. The information-feedback coding scheme in Sec. 3.4 uses only tail-biting convolutional codes.

Table 3.1, taken from Lin and Costello [17, Table 12.1], lists the generator polynomials for the rate-1/3 convolutional codes that were used as the mother codes for our simulations. Each code selected has the optimum free distance d_{free}, which is listed along with the analytic traceback depth L_D [40]. Higher-rate codewords used for the incremental transmissions are created by pseudorandom, rate-compatible puncturing of the rate-1/3 mother codes.

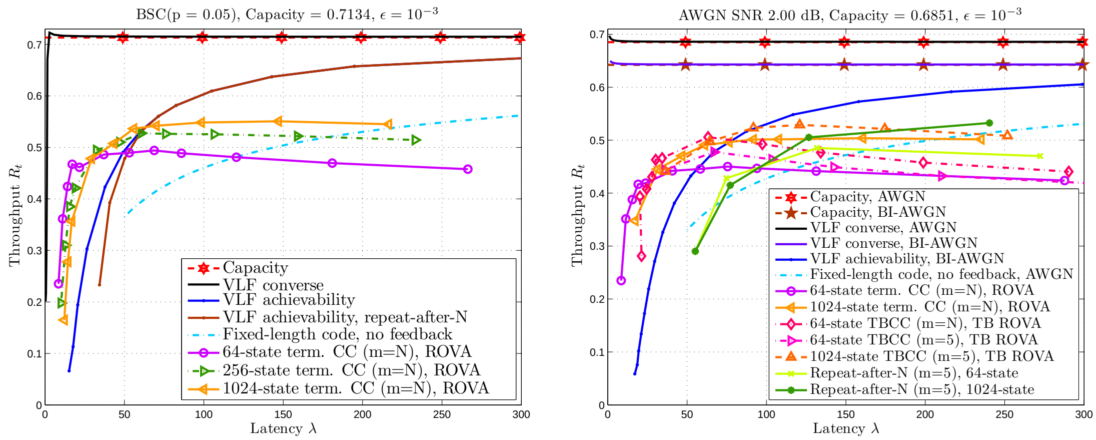
All of the simulations involving the AWGN channel use the binary-input AWGN channel (i.e., using BPSK signaling) with soft-decision decoding. The binary-input AWGN channel has a maximum Shannon capacity of 1 bit per channel use, even when the SNR η is unbounded. However, we have included comparisons with the capacity of the full AWGN channel (i.e., with real-valued inputs drawn i.i.d. $\sim \mathcal{N}(0, \eta)$). For the SNRs and capacities used in our examples, the binary-input restriction is a minor concern.

3.3.3 Numerical Results

Fig. 3.3(a) illustrates the short-blocklength performance of the reliability-based retransmission scheme using the ROVA for terminated convolutional codes, compared to the fundamental limits for VLF codes. This example uses the BSC with crossover probability $p = 0.05$ and target probability of error $\epsilon = 10^{-3}$. The Shannon (asymptotic) capacity of the BSC with crossover probability p is $C_{\text{BSC}} = 1 - h_{\text{b}}(p)$. The random-coding lower bound ('VLF achievability') is from Thm. 2 and the upper bound ('VLF converse') is from Thm. 6. An example of the random-coding lower bound for repeat-after-N codes from Cor. 1 is also shown ('VLF achievability, repeat-after-N'), with $N = 3 \log M$. Both the convolutional code simulations and the VLF bounds correspond to decoding after every received symbol. Fig. 3.3(a) also includes the maximum rate at finite blocklengths without feedback ('Fixed-length code, no feedback'), based on the normal approximation from [24].

Though the upper and lower bounds for VLF codes coincide asymptotically, there is a considerable gap when latency is below 100 bits, a region in which convolutional codes can deliver high rates. At the shortest blocklengths, the 64-state code with fewer memory elements performs best, due to the rate loss of the codes with larger constraint lengths. However, as the message size k increases (and the latency increases), the more powerful 1024-state code delivers superior throughput. As the latency continues to increase, the codes' throughputs fall below that of the VLF achievability bound, which is based on random coding. Random coding improves with latency, but the word-error performance of convolutional codes does not improve once the average latency is beyond twice the traceback depth L_D of the convolutional code [40].

The $(\lambda^{(\text{RB})}, R_t^{(\text{RB})})$ curve for the 64-state code exhibits non-monotonic behavior near $\lambda^{(\text{RB})}$=17 (k=8), likely due to non-monotonic minimum distance growth of the terminated convolutional codes as a function of blocklength, in conjunction with non-ideal effects of



(a) BSC (b) AWGN

Figure 3.3: Short-blocklength performance of the reliability-based retransmission scheme with target probability of error $\epsilon = 10^{-3}$. Simulations use the ROVA for terminated convolutional codes (term. CC) and the TB ROVA for tail-biting convolutional codes (TBCC), with decoding after every symbol (m=N) or with decoding after m=5 groups of symbols. Simulations are over (a) the BSC(0.05) or (b) over the AWGN channel with SNR 2.00 dB.

pseudo-random puncturing. The maximum throughput obtained for these BSC simulations is $R_t^{(\text{RB})} = 0.551$ bits per channel use at $\lambda^{(\text{RB})} = 147.06$ bits for the $k = 91$, 1024-state code, which is 77.2% of the BSC capacity.

Fig. 3.3(b) shows the performance of the reliability-based retransmission scheme over the AWGN channel with SNR 2 dB and target $\epsilon = 10^{-3}$. The Shannon capacity of the AWGN channel with SNR P is $C_{\text{AWGN}} = \frac{1}{2}\log(1+P)$, and the Shannon capacity of the BI-AWGN channel is computed using the approximations in [73]. The random-coding lower bound ('VLF achievability') is from Thm. 2, the AWGN upper bound ('VLF converse, AWGN') is from Thm. 5 and particularized to the Gaussian channel, and the BI-AWGN upper bound ('VLF converse, BI-AWGN') is from Thm. 6. Again, the "Fixed-length code, no feedback" curve uses the normal approximation [24].

The terminated convolutional code (term. CC) simulations in Fig. 3.3(b) use the ROVA and attempt decoding after every symbol, as in Fig. 3.3(a). Similar performance is observed.

The throughput of the convolutional codes surpasses the random-coding lower bound at short blocklengths, but plateaus around latencies of 100 bits. Convolutional codes with more memory elements would be expected deliver improved throughput, but computational complexity has limited us to codes with 1024 states. Additionally, the high decoding complexity of the 1024-state codes prevented us from decoding after every symbol using the TB ROVA.

3.3.4 Decoding after Groups of Symbols

In contrast, decoding less frequently is practically desirable due to the round-trip delay inherent in the feedback loop and because of the tremendous complexity associated with performing the ROVA after each received symbol. Decoding with the ROVA only after packets is a natural extension of the proposed scheme, akin to the m-transmission repeat-after-N_m codes in Sec. 3.2. When decoding only after packets are received, the latency $\lambda^{(\mathrm{RB})}$ and the throughput $R_t^{(\mathrm{RB})}$ become

$$\lambda^{(\mathrm{RB})} \leq \frac{I_1 + \sum_{i=1}^{m-1} I_i P_{\mathrm{NACK}}(N_i)}{1 - P_{\mathrm{NACK}}(N_m)}, \tag{3.60}$$

$$R_t^{(\mathrm{RB})} = \frac{k}{\lambda^{(\mathrm{RB})}}(1 - P_{\mathrm{UE}}). \tag{3.61}$$

Here $P_{\mathrm{NACK}}(N_i)$ is the probability of retransmission when N_i coded symbols have been received. The incremental transmission length at transmission i is I_i and the cumulative decoding blocklength is $N_i = I_1 + \cdots + I_i$.

A main challenge in an m-transmission incremental redundancy scheme is to select the set of m incremental transmission lengths $\{I_i\}_{i=1}^{m}$ that provide the best throughput at short blocklengths. In general, the retransmission probabilities in expressions such as (3.60) are non-convex and the blocklengths must be optimized numerically. Algorithm 1 in Sec. 3.6.7 presents a method to optimize the blocklengths in general incremental redundancy schemes. Sec. 3.6.8 describes how to particularize the algorithm in order to select the m=5 optimal blocklengths in the reliability-based retransmission scheme using the TB ROVA for TBCCs.

Based on the optimal transmission lengths identified by Algorithm 1, shown in Table 3.2, we simulated tail-biting convolutional codes (TBCCs) and the TB ROVA in an m=5 transmission decision-feedback scheme. Fig. 3.3(b) shows the impact on throughput when decoding is limited to these specified decoding opportunities. Despite fewer opportunities for decoding (and hence fewer chances to terminate transmission early), both the 64-state and 1024-state tail-biting codes in the optimized m=5 setting deliver excellent performance compared to the respective terminated codes that allow decoding after every symbol (i.e., m=N). Note also how at blocklengths less than approximately 75 bits, the m=5 TBCCs deliver higher rates than the random-coding lower bound that requires decoding after every symbol ('VLF achievability, BI-AWGN'). When compared to Thm. 2's random-coding lower bound for repeat-after-N_m codes on the AWGN channel ('Repeat-after-N (m=5)'), the m=5 TBCCs deliver higher rates for blocklengths up to about 125 bits. The 'Repeat-after-N (m=5), 64-state' curve uses the optimal m=5 blocklengths for the 64-state TBCC, and the 1024-state curve uses the optimal m=5 blocklengths for the 1024-state TBCC. The maximum throughput obtained from these m=5 simulations is $R_t^{(\mathrm{RB})} = 0.529$ bits per channel use at $\lambda^{(\mathrm{RB})} = 121.0$ bits, for the $k = 64$, 1024-state code. This is 77.2% of the AWGN capacity and 82.4% of the BI-AWGN capacity.

The performance of the tail-biting convolutional codes with only m=5 incremental transmissions is promising. Keep in mind, however, that the blocklengths were specifically optimized for one parameter set: $\epsilon = 10^{-3}$, SNR = 2 dB, fixed k, and a particular generator polynomial. Adapting the codes to channels with different SNRs or error requirements would require extensive re-characterization of the retransmission probabilities and optimization of the blocklengths. However, it is possible to use a heuristic choice of the m blocklengths that provides good throughput (if not optimal) across a limited range of SNRs.

Fig. 3.4 shows an example of 1024-state tail-biting convolutional codes simulated across a range of SNRs with the same choice of $m = 5$ blocklengths. In particular, for $k = 64$,

Table 3.2: Optimal transmission lengths $\{I_i\}^*$ for the m=5 transmission scheme using the tail-biting ROVA, for the AWGN channel with SNR $\eta = 2$ dB. The simulated error probability $P(\text{UE})$ corresponding to target error probability $\epsilon = 10^{-3}$ is also plotted.

		64-state TBCC	
Info. Bits k	Target Error ϵ	Transmission Lengths $\{I_1^*, I_2^*, I_3^*, I_4^*, I_5^*\}$	Simulated Error $P(\text{UE})$
16	10^{-3}	30, 3, 3, 5, 7	2.260×10^{-4}
32	10^{-3}	57, 6, 7, 9, 16	1.960×10^{-4}
48	10^{-3}	88, 9, 10, 13, 24	2.350×10^{-4}
64	10^{-3}	121, 12, 13, 17, 29	2.600×10^{-4}
91	10^{-3}	178, 17, 18, 22, 38	2.650×10^{-4}
128	10^{-3}	261, 23, 24, 30, 46	2.440×10^{-4}

		1024-state TBCC	
Info. Bits k	Target Error ϵ	Transmission Lengths $\{I_1^*, I_2^*, I_3^*, I_4^*, I_5^*\}$	Simulated Error $P(\text{UE})$
16	10^{-3}	29, 4, 4, 4, 7	2.473×10^{-4}
32	10^{-3}	56, 5, 5, 7, 12	1.976×10^{-4}
48	10^{-3}	80, 7, 7, 9, 16	2.085×10^{-4}
64	10^{-3}	106, 9, 9, 12, 22	1.993×10^{-4}
91	10^{-3}	151, 13, 14, 17, 31	2.197×10^{-4}
128	10^{-3}	223, 17, 18, 24, 44	2.344×10^{-4}

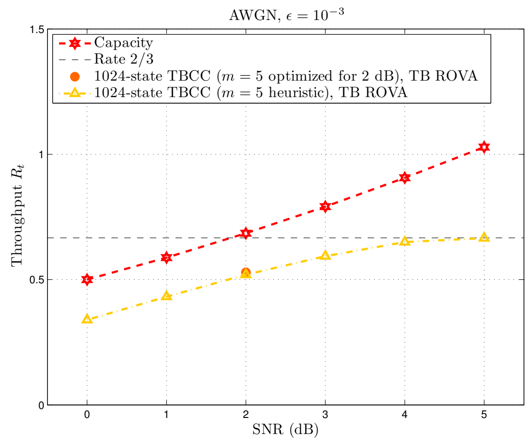


Figure 3.4: Performance of a heuristic blocklength-selection policy for $\{I_i\}_{i=1}^5$ used across a range of SNRs for a 1024-state tail-biting convolutional code with $k = 64$ message bits. For all points, the target probability of error is $\epsilon = 10^{-3}$.

the $m = 5$ blocklengths were $\{I_1 = \frac{3}{2}k = 96, I_2 = \frac{1}{4}k = 16, I_3 = \frac{1}{4}k = 16, I_4 = \frac{1}{4}k = 16, I_5 = \frac{3}{4}k = 48\}$. The corresponding 2 dB throughput for the optimized $m = 5$, $k = 64$ blocklengths from Fig. 3.3(b) is also shown. For reference, the throughput corresponding to the optimized blocklengths for the 1024-state code is 0.5290, whereas the throughput for the heuristic blocklengths is 0.5196. Fig. 3.4 demonstrates that as the SNR increases, the maximum rate achieved by this heuristic blocklength-selection policy is $\frac{2}{3}$, since the highest rate possible is $\frac{k}{I_1} = \frac{2}{3}$. A more aggressive (higher rate) initial length I_1 should be chosen if SNRs above 4 dB are expected, which may reduce the throughput at low SNRs.

Note that for all of the VLF simulations shown in this section, at least 25 undetected word-errors were accumulated for each value of k. Because the ROVA-based stopping rule with target $\epsilon = 10^{-3}$ guarantees that the average probability of error is no more than 10^{-3}, it is not important for us to characterize the error probability exactly. For this chapter, we are more concerned with the performance in terms of throughput and latency. Observations from VLF simulations show that collecting at least 25 word errors is sufficient for estimating the throughput and latency.

Table 3.3: Simulated probabilities of undetected error $P(\text{UE})$ corresponding to the CRC simulations in Fig. 3.5 for the 2 dB AWGN channel. The 12-bit CRCs fail to meet the error constraint of $\epsilon = 10^{-3}$, but the stronger 16-bit CRCs generally satisfy the constraint. Generator polynomials for "good" A-bit CRCs from [4] are listed in hexadecimal in parentheses. For example, 0xcd indicates the polynomial $x^8 + x^7 + x^4 + x^3 + x + 1$. The $k + A$ column indicates the number of bits input to the rate-1/3 convolutional encoder. Cells labeled – indicate that no simulations were run for that value of $k + A$.

Input Bits $k + A$	$P(\text{UE})$ 12-bit CRC (0xc07)	$P(\text{UE})$ 16-bit CRC (0x8810)
24	1.479×10^{-1}	1.000×10^{-4}
30	–	8.618×10^{-4}
34	1.036×10^{-3}	1.077×10^{-3}
40	–	2.105×10^{-4}
48	2.935×10^{-3}	2.025×10^{-4}
64	3.504×10^{-3}	2.538×10^{-4}
91	4.646×10^{-3}	3.017×10^{-4}
128	6.755×10^{-3}	5.370×10^{-4}
181	8.864×10^{-3}	6.667×10^{-4}

3.3.5 Code-based Error Detection

In practice, decision-feedback schemes often use a checksum (e.g., a CRC) at the receiver to detect errors in the decoded word. However, the additional parity bits of a CRC that must be sent impose a latency cost that may be severe at short blocklengths. For an A-bit CRC appended to k message bits, the throughput (not counting the check bits) is $R_t^{(\text{CRC})} = \frac{k}{k+A} R_t$, where R_t is defined similarly to (3.54) for the reliability-based scheme. Equivalently, the rate-loss factor from an A-bit CRC is $\frac{A}{k+A}$.

Using a error-detection code to determine retransmission requests is sometimes referred to as code-based error detection [15], in contrast to reliability-based error detection with the ROVA. As noted in Frick and Hoeher's [15] investigation of reliability-based hybrid ARQ schemes, the rate loss and undetected error probability of the code-based approach depend critically on the blocklengths and target error probabilities involved.

Fig. 3.5 provides an example of the throughput obtained when decoding after every symbol and using a 16-bit CRC for error detection. After decoding the 64-state TBCC (which has $k + A$ input bits), the receiver re-computes the CRC to check for errors. As expected, the rate loss of the CRCs at blocklengths less than 50 bits severely limits the achievable rates. The 64-state TBCCs decoded with the TB ROVA deliver higher rates until the average blocklength reaches about 75 to 100 bits. For moderately large blocklengths (e.g., 150 bits and greater), the throughput penalty induced by CRCs becomes less severe. (As the information length k increases, the rate-loss factor $\frac{A}{k+A}$ decays to zero.) Note that decoding after every symbol prevents simulation of higher-constraint-length convolutional codes (e.g., 1024-state codes).

Importantly, using ROVA guarantees the probability of error to be less than ϵ (as long as the length-N mother code is long enough to meet this constraint on average), but CRCs provide no such guarantee. In this example, in fact, TBCC simulations with 12-bit CRCs failed to meet the target of $\epsilon = 10^{-3}$, as shown in Table 3.3. As a result, the codes with 12-bit CRCs do not qualify as $(\ell, M = 2^k, \epsilon = 10^{-3})$ VLF codes and the throughput obtained for the 12-bit CRCs is not plotted in Fig. 3.5. Both the 12-bit and 16-bit CRC polynomials are from [4] and are listed in Table 3.3.

The 16-bit CRCs generally provide sufficiently low error probabilities, but at the expense of reduced rate versus the 12-bit CRCs. One exception when the 16-bit CRC fails to meet the $\epsilon = 10^{-3}$ constraint in our simulations is when $k + A = 34$ ($k \approx A = 16$), as shown in Table 3.3. This high error probability seems to be an outlier compared to the other 16-bit CRC simulations, but is is consistent with findings from previous CRC research, such

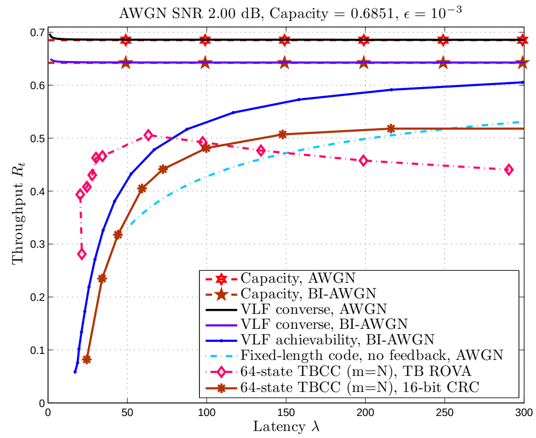


Figure 3.5: A comparison of two decision-feedback schemes, the reliability-based retransmission scheme with the ROVA and the code-based approach with 12-bit and 16-bit CRCs. Using ROVA guarantees the probability of error to be less than ϵ, but CRCs provide no such guarantee. In this example, the 12-bit CRCs fail to meet the target of $\epsilon = 10^{-3}$.

as [74]. In [74], Witzke and Leung show that the undetected error probability of a given A-bit CRC polynomial over the BSC can vary widely as a function of the channel bit-error rate, especially for small values of k. The present variable-length simulations complicate matters further, due to the stopping rule that terminates when the CRC matches, even if erroneously. Additional simulations with $k + A = 30$ and $k + A = 40$ were performed in order to illustrate the sensitivity of the error probability to the information length k (Table 3.3).

In general, it is difficult to determine exactly how many CRC bits should be used to provide the maximum throughput for hybrid ARQ schemes, since the error probability depends on the SNR. Communication system designers will likely be tempted to conservatively select large CRC lengths, which restricts the short-blocklength throughput. In contrast, the ROVA-based approach always guarantees a target error probability. Still, future work that explores the performance of VLF codes at larger blocklengths (e.g., ~ 400 bits and above) may benefit from code-based error detection.

3.4 Information-feedback Codes

3.4.1 Two-Phase Incremental Redundancy

In the previous section, we demonstrated that convolutional codes decoded with the ROVA in a decision-feedback setting could deliver throughput above the random-coding lower bound. Does using information feedback allow us to achieve even greater rates? This section explores this question with a two-phase incremental redundancy scheme from [56] that uses information feedback and improves upon the short-blocklength performance of the ROVA-based scheme. Similar to the two-phase schemes used in Thms. 3 and 4, the proposed scheme consists of communication and confirmation phases. Here both phases have fixed lengths.

In the communication phase, after the receiver attempts to decode a block of coded symbols, noiseless feedback informs the transmitter of the decoding result. In the subsequent confirmation phase, the transmitter sends a coded ACK/NACK on the forward channel depending on whether decoding was successful. Via noiseless feedback, the receiver informs the transmitter which confirmation message (ACK or NACK) it decoded. If the receiver decodes an ACK, it will proceed to the next message. If the receiver decodes a NACK, the two phases are repeated until an ACK is decoded at the receiver in the confirmation phase. A main difference in the proposed scheme compared to Thms. 3 and 4 is that in this case, both phases have fixed lengths. Instead of the receiver stopping transmission when its estimate is sufficiently reliable, as in the decision-feedback schemes of Sec. 3.3, the transmission lengths are chosen to be sufficiently large to guarantee the ϵ error constraint. A practical implication is that feedback need not be sent until the end of each fixed-length phase.

Fig. 3.6 shows a diagram of the proposed two-phase incremental redundancy scheme. I_1 symbols are transmitted in the first communication phase and decoded with blocklength $N_1 = I_1$. Next, N_{conf} symbols conveying ACK or NACK are transmitted in the confirmation phase. The total number of symbols sent in the ith two-phase cycle is $I_i' = I_i + N_{\text{conf}}$. Transmission stops when the receiver has decoded an ACK in the confirmation phase, even if the

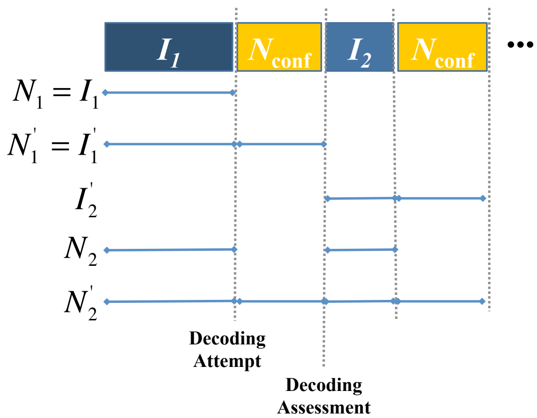


Figure 3.6: A notional diagram of the two-phase communication scheme. The ith two-phase cycle ($i = 1, \ldots, m$) has a communication-phase transmission with I_i symbols and a confirmation-phase transmission with N_{conf} symbols. The number of symbols transmitted in the ith cycles is $I_i' = I_i + N_{\text{conf}}$. N_i is the number of communication-phase symbols transmitted by the end of the ith cycle. N_i' is the total number of symbols transmitted by the end of the ith cycle. The dashed vertical lines indicate the communication phase decoding attempts and confirmation phase ACK/NACK decoding assessments. Feedback occurs at each of these vertical lines.

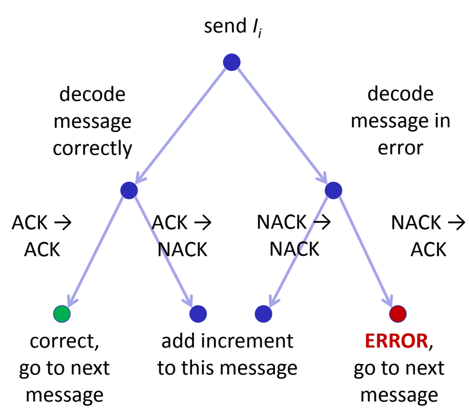


Figure 3.7: An illustration of the events that can lead to correct decoding, undetected errors, or additional retransmissions.

transmitter actually sent a NACK. If a NACK is decoded, another round of communication and confirmation occurs; I_2 symbols are transmitted and decoded using the cumulative blocklength $N_2 = N_1 + I_2$, and then another N_{conf} symbols are transmitted.

The communication-phase transmission length in the ith transmission is I_i and the decoding blocklength is $N_i = N_{i-1} + I_i$. The confirmation-phase transmission length is always N_{conf}. The ith confirmation message is decoded independently of the previous confirmation blocks. If the receiver still has not decoded an ACK by the mth transmission, the scheme disregards all earlier transmissions and starts over with I_1 communication symbols and N_{conf} confirmation symbols. The total number of channel uses at the ith decoding attempt, N'_i, is

$$N'_i = \begin{cases} \sum\limits_{j=1}^{i} I'_j \,, & 1 \le i \le m \\ uN'_m + \sum\limits_{j=1}^{v} I'_j \,, & i = um + v \end{cases} . \tag{3.62}$$

The scheme is analogous to type-I hybrid ARQ with an additional confirmation phase when the maximum number of transmissions m is 1.

As shown in Fig. 3.7, message decoding errors occur only when a forward NACK is decoded as an ACK (occurring with probability $p_{n\to a} = 1 - p_{n\to n}$), in which case the receiver is unaware that it has decoded incorrectly. (All message errors are undetected errors.) When forward ACKs are decoded as NACKs (occurring with probability $p_{a\to n} = 1 - p_{a\to a}$), the message incurs additional latency due to retransmission, but no errors are made.

We denote the probability of decoding incorrectly in decoding attempt i with blocklength N_i as $P(\zeta_i)$ and the probability that the decoder picks the correct codeword as $P(\zeta_i^c) = 1 - P(\zeta_i)$. We assume that successful decoding of confirmation messages occurs with equal probability regardless of whether ACK or NACK was sent, i.e., $p_{a\to a} = p_{n\to n}$.

Throughout this section, we assume that a simple repetition code of length N_{conf} is sent in the confirmation phase. For the binary-input Gaussian channel with SNR η, this means the confirmation-phase error probability is $p_{n\to a} = Q(\sqrt{N_{\text{conf}}\eta})$, where $Q(u) = \frac{1}{\sqrt{2\pi}} \int_u^\infty \exp \frac{-t^2}{2} dt$ is the tail probability of the standard normal distribution. For the BSC with crossover probability p, the confirmation-phase error probability is

$$p_{n\to a} = \begin{cases} \sum\limits_{t=\lceil N_{\text{conf}}/2\rceil}^{N_{\text{conf}}} \binom{N_{\text{conf}}}{t} p^t (1-p)^{N_{\text{conf}}-t}, & N_{\text{conf}} \text{ odd} \\ \frac{1}{2}\binom{N_{\text{conf}}}{N_{\text{conf}}/2} p^{N_{\text{conf}}/2}(1-p)^{N_{\text{conf}}/2} + \sum\limits_{t=N_{\text{conf}}/2+1}^{N_{\text{conf}}} \binom{N_{\text{conf}}}{t} p^t (1-p)^{N_{\text{conf}}-t}, & N_{\text{conf}} \text{ even} \end{cases} \tag{3.63}$$

Let $\mathcal{N}_i$ be the event that the receiver decodes the ith confirmation message as a NACK, which has the following probability:

$$P(\mathcal{N}_i) = \begin{cases} P(\zeta_1)p_{n\to n} + P(\zeta_1^c)p_{a\to n}, & i=1 \\ P(\mathcal{N}_{i-1})P(\zeta_i|\mathcal{N}_{i-1})p_{n\to n} + P(\mathcal{N}_{i-1})P(\zeta_i^c|\mathcal{N}_{i-1})p_{a\to n}, & 2\le i\le m \\ P(\mathcal{N}_m)^u P(\mathcal{N}_v), & i = um+v \end{cases}. \tag{3.64}$$

For convenience we also define $P(\mathcal{N}_0) = 1$. With probability $P(\mathcal{N}_i)$, the transmitter sends increment $(i+1)$ with length I_{i+1}, followed by another N_{conf} confirmation symbols.

For $2 \le i \le m$, the expression for $P(\mathcal{N}_i)$ in (3.64) can be expanded into terms similar to those for the $i = 1$ case. Because of the dependence on previous decoding outcomes, the number of these terms grows exponentially with i, but the overall probability is tightly upper bounded by the two dominant terms as follows:

$$P(\mathcal{N}_i) < P(\zeta_1,\zeta_2,\ldots,\zeta_i)p_{n\to n}^i + P(\zeta_1,\zeta_2,\ldots,\zeta_{i-1},\zeta_i^c)p_{n\to n}^{i-1}p_{a\to n}\,. \tag{3.65}$$

Approximating the joint error probability as the marginal error probability, i.e.,
$P(\zeta_1,\zeta_2,\ldots,\zeta_i) \approx P(\zeta_i)$, we have the following approximation for $2 \le i \le m$:

$$P(\mathcal{N}_i) \approx P(\zeta_i)p_{n\to n}^i + [P(\zeta_{i-1}) - P(\zeta_i)]p_{n\to n}^{i-1}p_{a\to n}\,. \tag{3.66}$$

Similarly, the probability of undetected error in the ith transmission, $P(\text{UE}_i)$, is well approximated as $P(\text{UE}_i) \approx P(\zeta_i)p_{n\to n}^{i-1}p_{n\to a}$ for $i = 1,\ldots,m$. For integers $u \ge 1$ and $1 \le v \le m$, $P(\text{UE}_{um+v}) = P(\mathcal{N}_m)^u P(\text{UE}_v)$. The overall probability of (undetected) error for a message is

$$P(\text{UE}) = (1 - P(\mathcal{N}_m))^{-1}\, P(\text{UE}_1^m), \tag{3.67}$$

where

$$P(\mathrm{UE}_1^m) = \sum_{i=1}^{m} P(\mathrm{UE}_i). \tag{3.68}$$

Note that unlike the two-phase achievability schemes in Thms. 3 and 4, the receiver in this scheme retains previous received symbols after decoding a NACK in the confirmation phase, until the mth two-phase cycle. This dependence complicates the analysis and leads to the approximations in (3.66). In contrast, the receiver in Thms. 3 and 4 discards all received symbols and starts over whenever it decodes a NACK in the confirmation phase. As a result, no approximations are required.

With these equations for the error probability and the probability of decoding a NACK, we can compute the expected blocklength and rate of the two-phase scheme. The expected latency $\lambda^{\text{(two-phase)}}$ (i.e., the average number of channel uses before decoding an ACK at the receiver) and throughput $R_t^{\text{(two-phase)}}$ are computed as follows:

$$\lambda^{\text{(two-phase)}} = (1 - P(\mathcal{N}_m))^{-1} \sum_{i=1}^{m} I'_i P(\mathcal{N}_{i-1}), \tag{3.69}$$

$$R_t^{\text{(two-phase)}} = \frac{k}{\lambda^{\text{(two-phase)}}}(1 - P(\mathrm{UE})) \tag{3.70}$$

$$= \frac{k(1 - P(\mathcal{N}_m))(1 - P(\mathrm{UE}))}{\sum\limits_{i=1}^{m} I'_i P(\mathcal{N}_{i-1})} \tag{3.71}$$

$$= \frac{k(1 - P(\mathcal{N}_m) - P(\mathrm{UE}_1^m))}{\sum\limits_{i=1}^{m} I'_i P(\mathcal{N}_{i-1})}, \tag{3.72}$$

where k is the number of information symbols in each attempted message. The expression in (3.69) includes the factor $(1-P(\mathrm{UE}))$ so that $R_t^{\text{(two-phase)}}$ excludes undetected errors and thus only counts messages that are decoded successfully at the receiver. Note the instantaneous rate at the completion of the ith transmission is $R_i = k/N'_i$. Finally, the expected number of decoding attempts $D^{(\text{two-phase})}$ is:

$$D^{(\text{two-phase})} = (1 - P(\mathcal{N}_m))^{-1} \sum_{i=1}^{m} P(\mathcal{N}_{i-1})\,. \tag{3.73}$$

3.4.2 Rate-Compatible Sphere-Packing Analysis

The expressions (3.69-3.73) are general and may be applied to any error-correction code and any channel model. In this section, we use the rate-compatible sphere-packing (RCSP) analysis of Williamson et al. [55] and Chen et al. [50, 75] to approximate the two-phase performance possible for an idealized rate-compatible family of sphere-packing codes. RCSP assumes that the code achieves a (geometrically impossible) perfect packing of decoding spheres at each of the jth communication-phase transmissions ($j = 1, \ldots, m$). A discussion of the resulting error probability for both bounded-distance (BD) and maximum-likelihood (ML) decoders follows.

Example. For the AWGN with SNR η, the squared sphere-packing decoding radius corresponding to blocklength N_j is $r_j^2 = N_j(1+\eta)\ 2^{-2k/N_j}$. For a BD decoder, errors occur when the noise power is larger than the squared decoding radius. The sphere-packing (SP) probability of decoding error $\mathrm{P}_{\mathrm{SP}}^{\mathrm{BD}}(\zeta_j)$ associated with radius r_j is

$$\mathrm{P}_{\mathrm{SP}}^{\mathrm{BD}}(\zeta_j) = P\left(\sum_{n=1}^{N_j} z_n^2 > r_j^2\right) = 1 - F_{\chi^2_{N_j}}(r_j^2),\ \text{(AWGN)} \tag{3.74}$$

where the noise samples are $z_n \sim \mathcal{N}(0,1)$ and $F_{\chi^2_{N_j}}(u)$ is the CDF of a chi-square random variable with N_j degrees of freedom. If an ML decoder is used instead of a BD decoder, (3.74) is not a bound on the error performance, but can provide a close approximation in some cases for convolutional codes [55]. Note that (3.74) is the marginal probability of error, which does not depend on decoding success in the $(j-1)$th decoding attempt.

Shannon's sphere-packing bound [76] provides a lower bounds on the optimal probability of error for an ML decoder. In this chapter, we use the asymptotic approximation of (51) in [76]. As described in Chen et al. [50], Shannon's asymptotic approximation is not meaningful for rates above capacity, so we have used the RCSP BD error probability in (3.74) for these rates to provide a meaningful lower bound. □

Example. For the BSC with crossover probability p, the probability of error for BD decoding with blocklength N_j can be lower bounded as follows [50]:

$$\mathrm{P}_{\mathrm{SP}}^{\mathrm{BD}}(\zeta_j) \geq \sum_{t=r_j+1}^{N_j} \binom{N_j}{t} p^t (1-p)^{N_j - t}, \text{ (BSC)} \tag{3.75}$$

where r_j is the radius of the smallest N_j-dimensional sphere that contains at least 2^{N_j-k} words:

$$r_j = \left\{ r : \sum_{t=0}^{r-1} \binom{N_j}{t} < 2^{N_j-k}, \sum_{t=0}^{r} \binom{N_j}{t} \geq 2^{N_j-k} \right\}. \tag{3.76}$$

For all blocklengths $N_j < k$, the radius r_j is zero and the probability of error $\mathrm{P}_{\mathrm{SP}}^{\mathrm{BD}}(\zeta_j)$ is exactly one. As noted in Chen et al. [50], when the 2^k uniform decoding regions perfectly fill the N_j-dimensional space, ML decoding and BD decoding are identical.

For a fixed message length k, the lower bound on error probability given in (3.75) is not monotone in the blocklength N_j, which makes it difficult to optimize the blocklengths $\{N_i\}$. In the optimizations that follow, we use the following approximation instead:

$$\mathrm{P}_{\mathrm{SP}}^{\mathrm{BD}}(\zeta_j) \approx (1-\rho_j)\binom{N_j}{r_j} p^{r_j}(1-p)^{N_j-r_j} + \sum_{t=r_j+1}^{N_j} \binom{N_j}{t} p^t (1-p)^{N_j-t} \text{ (BSC)} \tag{3.77}$$

$$\rho_j = \frac{2^{N_j-k} - \sum_{t=0}^{r_j-1} \binom{N_j}{t}}{\binom{N_j}{r_j}}. \tag{3.78}$$

That is, ρ_j is the fraction of the N_j-dimensional words at radius r_j from a particular codeword that "belong" to that codeword. For this approximate BD decoder, an error is declared if the noise causes the received word to lie outside the sphere of radius r_j, or, with probability $(1-\rho_j)$, if the received word lies exactly at distance r_j from the transmitted codeword. □

3.4.3 Optimization of Blocklengths Using RCSP

Eqs. (3.74-3.76) provide bounds on the communication-phase error probabilities for blocklength N_j in the proposed two-phase scheme, but it remains to select the optimal blocklength for each of the m transmissions ($j \in \{1, \dots, m\}$). In particular, we would like to select the transmission lengths $\{I_i\}$ and N_{conf} that maximize the throughput $R_t^{(\text{two-phase})}$, under a constraint on the undetected error probability $P(\text{UE})$, as follows:

$$\{I_i\}^*, N_{\text{conf}}^* = \arg \max_{\{I_i\}, N_{\text{conf}}} R_t^{(\text{two-phase})} \quad \text{s.t.} \quad P(\text{UE}) \leq \epsilon. \tag{3.79}$$

Even for relatively small m (e.g., $m = 3$), this task is non-trivial. Algorithm 1 in Appendix 3.6.7 describes an efficient method for computing locally-optimal blocklengths.

Table 3.4 lists the optimal blocklengths identified by Algorithm 1 in an m=5 two-phase scheme for the AWGN channel with SNR $\eta = 2.00$ dB, for $k \in \{16, 32, 64, 91, 128\}$, using the constraint that $\epsilon \leq 10^{-3}$. Table 3.4 assumes probabilities of error predicted by RCSP with BD decoding, as in (3.74). Similarly, Table 3.5 lists the optimal two-phase blocklengths identified by Algorithm 1 in an m=5 two-phase scheme for the BSC($p = 0.05$), for $k \in \{16, 32, 64, 91, 128\}$, using the constraint that $\epsilon \leq 10^{-3}$. Table 3.5 assumes probabilities of error predicted by RCSP with BD decoding, as in (3.77).

Figs. 3.8 and 3.9 illustrates the corresponding rates that would be achievable with idealized RCSP code families, for the AWGN channel and BSC, respectively. Each point on the 'Two-phase RCSP' curve represents the throughput and latency corresponding to the optimal m=5 communication-phase and confirmation-phase blocklengths (determined by Algorithm 1) for a fixed message size k. For the AWGN channel, the results of the optimization for both the ML and BD versions of the RCSP analysis are shown.

Interestingly, we observed that increasing the number of decoding opportunities m did not significantly increase the throughput predicted by the RCSP BD analysis for the AWGN channel. This contrasts with the VLFT setting of [55], in which larger m values were associated with improved rates. In this two-phase scheme, increasing m incurs a latency overhead of N_{conf} additional symbols, which explains why the rate does not necessarily improve with increased m.

3.4.4 Two-Phase Convolutional Code Simulations, AWGN

This section shows that the two-phase RCSP analysis can closely predict the performance of simulated tail-biting convolutional codes (TBCCs) and, more importantly, that TBCCs can deliver throughput surpassing both the random-coding lower bound and Sec. 3.3's ROVA-based decision-feedback scheme. In particular, the simulated m=5 two-phase scheme used TBCCs in the communication phase and a repetition code of length N_{conf} in the confirmation phase. The 64-state and 1024-state convolutional code polynomials are specified in Table 3.1.

The blocklengths used for the TBCCs, given in Table 3.4, are those identified by the blocklength-selection algorithm that assumes RCSP. Also shown in Table 3.4 are the simulated probabilities of undetected error $P(\text{UE})$ corresponding to both 64-state and 1024-state implementations of the TBCCs.

For some values of k, the simulated error probabilities exceeded the target error probability of $\epsilon = 10^{-3}$, which means that the listed set of convolutional code blocklengths does not qualify as an $(\ell, M = 2^k, \epsilon = 10^{-3})$ VLF code. This is not unexpected, since not all tail-biting convolutional codes can match the idealized RCSP BD approximation. This deficiency is particularly noticeable for larger values of k, since the word-error probability of convolutional codes decays once the blocklength is several times the traceback depth [40].

In order to fairly compare the throughput obtained for simulated codes with the VLF bounds from Sec. 3.2, we must use codes that satisfy the ϵ error constraint. Table 3.4 also provides the optimal blocklengths for the two-phase scheme using stricter error targets of 3×10^{-4} and 10^{-4}, followed by the simulated error probabilities $P(\text{UE})$ corresponding to these tail-biting convolutional codes.

Finally, using only the sets of blocklengths from Table 3.4 that meet the error constraint

Table 3.4: Optimal transmission lengths $\{I_i\}^*$ and N^*_{conf} obtained from the blocklength-selection algorithm for the two-phase scheme on the AWGN channel, assuming RCSP error probabilities for bounded-distance decoding. The simulated error probability $P(\text{UE})$ for the two-phase scheme with 64-state and 1024-state TBCCs is also shown. Only simulations achieving $P(\text{UE}) \leq \epsilon = 10^{-3}$ are shown in Fig.3.8. Both RCSP analysis and TBCC simulations use m=5 transmissions and SNR $\eta = 2$ dB. $P(\text{UE})$ values listed as $-$ were not recorded since larger values of the target error ϵ yielded satisfactory results.

Info. Bits k	Transmission Lengths $\{I^*_1, I^*_2, I^*_3, I^*_4, I^*_5, N^*_{\text{conf}}\}$	Target Error ϵ	Simulated $P(\text{UE})$ 64-state TBCC	Simulated $P(\text{UE})$ 1024-state TBCC
16	27, 7, 6, 6, 9, 5	10^{-3}	6.150×10^{-4}	6.239×10^{-4}
32	51, 11, 11, 11, 12, 5	10^{-3}	8.780×10^{-4}	6.400×10^{-4}
64	100, 15, 11, 10, 18, 5	10^{-3}	1.881×10^{-3}	9.667×10^{-4}
91	131, 15, 13, 14, 23, 6	10^{-3}	1.846×10^{-3}	9.314×10^{-4}
128	183, 17, 15, 16, 28, 6	10^{-3}	2.753×10^{-3}	1.674×10^{-3}
16	28, 10, 7, 6, 9, 6	3×10^{-4}	1.920×10^{-4}	2.155×10^{-4}
32	54, 11, 8, 8, 12, 6	3×10^{-4}	2.230×10^{-4}	$-$
64	95, 13, 11, 12, 19, 7	3×10^{-4}	4.580×10^{-4}	$-$
91	134, 15, 13, 15, 22, 7	3×10^{-4}	6.720×10^{-4}	$-$
128	188, 18, 17, 16, 27, 7	3×10^{-4}	1.016×10^{-3}	6.667×10^{-4}
16	29, 12, 9, 7, 9, 7	10^{-4}	5.500×10^{-5}	$-$
32	50, 10, 9, 9, 13, 8	10^{-4}	7.900×10^{-5}	$-$
64	97, 14, 11, 12, 18, 8	10^{-4}	1.790×10^{-4}	$-$
91	137, 17, 13, 13, 22, 8	10^{-4}	2.760×10^{-4}	$-$
128	191, 22, 16, 15, 26, 8	10^{-4}	3.670×10^{-4}	$-$

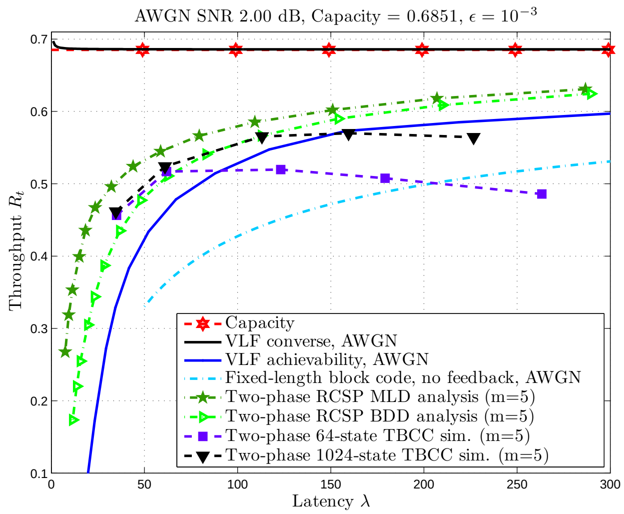


Figure 3.8: Achievable rates of the m=5 two-phase incremental redundancy scheme for the AWGN channel, including predictions from RCSP analysis and simulations using tail-biting convolutional codes (TBCCs). The AWGN channel SNR is 2 dB and the undetected error probability is constrained to be less than $\epsilon = 10^{-3}$.

of $P(\text{UE}) \leq 10^{-3}$, Fig. 3.8 displays the throughput results of the tail-biting convolutional code simulations. For the values of k where more than one target error provided a simulated $P(\text{UE}) \leq 10^{-3}$, only the row with the greatest $P(\text{UE})$ is displayed in Fig. 3.8. (This is also the row with the greatest throughput.)

Fig. 3.8 includes the fundamental limits for VLF codes from Sec. 3.2. The 'VLF converse' is from Thm. 5 and the 'VLF achievability' curve is from the random-coding lower bound of Thm. 2, both originally from Polyanskiy et al. [25]. Fig. 3.8 demonstrates that convolutional codes in the two-phase scheme can deliver throughput at least as high as the VLF random-coding lower bound at low latencies. This is true for the m=5 two-phase scheme despite the fact that the random-coding bound allows the decoder to terminate after transmission of any individual symbol. Finally, the 'Fixed-length code, no feedback' curve uses the normal approximation from [24]. Note that even though multiple achievability theorems in [24] predict rates close to both the converse and the normal approximation, results for explicit codes that achieve the non-asymptotic optimal rates are scarce.

At the shortest blocklengths ($k \in \{16, 32\}$), Fig. 3.8 shows that ML-decoded convolutional codes provide slightly better error performance than the RCSP analysis based on bounded-distance decoding, which results in superior throughput for the convolutional codes. As k increases, however, the convolutional code performance falls short of the RCSP prediction. Eventually, around a latency of 150 bits, even the 1024-state convolutional code's performance lags that of the VLF lower bound. The maximum throughput obtained from these simulations is $R_t^{(\text{two-phase})} = 0.5699$ bits per channel use at $\lambda^{(\text{two-phase})} = 159.5$ bits for the $k = 91$, 1024-state code, which is 83.2% of the AWGN capacity. The poor performance of the convolutional codes for larger values of k is expected because the free distance of the convolutional codes does not improve once the blocklength of the mother code has exceeded the analytic traceback depth L_D [40]. In contrast, the RCSP analysis and the VLF random-coding lower bound both have code performance that continues to improve as blocklength increases.

3.4.5 Two-Phase Convolutional Code Simulations, BSC

Table 3.5 lists the two-phase blocklengths identified by Algorithm 1 that were used for the TBCC simulations over the binary symmetric channel with crossover probability $p = 0.05$. Table 3.5 also gives the simulated probabilities of undetected error $P(\text{UE})$ corresponding to both 64-state and 1024-state implementations of the two-phase scheme with tail-biting convolutional codes.

Fig. 3.9 displays the corresponding throughput results of the TBCC simulations. Fig. 3.9 includes the VLF bounds on the maximum rate from Sec. 3.2. The 'VLF converse' is from Thm. 6 and the 'VLF achievability' curve is from random-coding lower bound of Thm. 2, both originally from Polyanskiy et al. [25]. The 'Fixed-length code, no feedback' curve uses the normal approximation [24].

Similarly to the AWGN example, Fig. 3.9 demonstrates that the convolutionally coded

Table 3.5: Optimal transmission lengths $\{I_i\}^*$ and N^*_{conf} obtained from the blocklength-selection algorithm for the two-phase scheme over the BSC, assuming RCSP error probabilities. The simulated error probability $P(\text{UE})$ for the two-phase scheme with 64-state and 1024-state TBCCs is also shown. Fig.3.9 shows the corresponding throughputs. Both RCSP analysis and TBCC simulations use $m = 5$ transmissions and crossover probability $p = 0.05$.

Info. Bits	Transmission Lengths	Target	Simulated $P(\text{UE})$	
k	$\{I^*_1, I^*_2, I^*_3, I^*_4, I^*_5, N^*_{\text{conf}}\}$	Error ϵ	64-state	1024-state
16	$25, 8, 8, 3, 6, 3$	10^{-3}	6.620×10^{-4}	4.368×10^{-4}
32	$42, 8, 8, 7, 10, 5$	10^{-3}	1.000×10^{-4}	7.036×10^{-5}
64	$85, 9, 8, 11, 15, 5$	10^{-3}	1.970×10^{-4}	1.141×10^{-4}
91	$119, 13, 12, 12, 19, 5$	10^{-3}	2.270×10^{-4}	1.260×10^{-4}
128	$172, 13, 9, 16, 23, 5$	10^{-3}	3.740×10^{-4}	2.078×10^{-4}

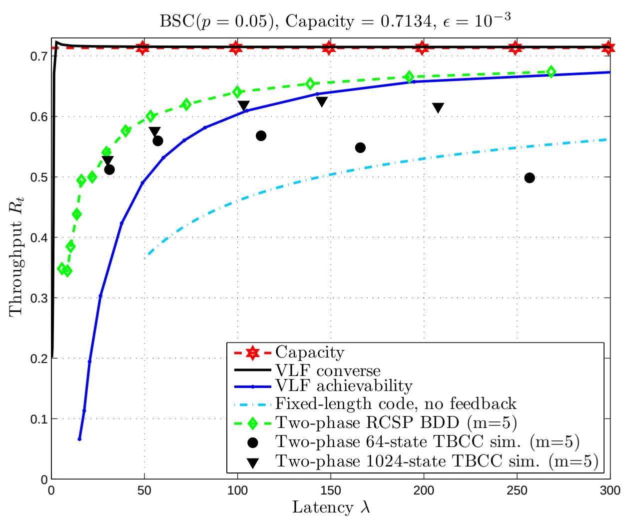


Figure 3.9: Achievable rates of the $m = 5$ two-phase incremental redundancy scheme for the BSC, including predictions from RCSP analysis using bounded-distance decoding (BDD) and simulations using tail-biting convolutional codes (TBCC sim.). The BSC crossover probability is $p = 0.05$ and the undetected error probability is constrained to be less than $\epsilon = 10^{-3}$.

two-phase scheme over the BSC can deliver throughput at least as high as the VLF random-coding lower bound at low latencies. This is true for the m=5 two-phase scheme despite the fact that the random-coding bound allows the decoder to terminate after transmission of any individual symbol. The maximum throughput obtained from these simulations is $R_t^{(\text{two-phase})} = 0.6265$ bits per channel use at $\lambda^{(\text{two-phase})} = 145.2$ bits for the $k = 91$, 1024-state code, which is 87.8% of the BSC capacity.

3.4.6 Information-feedback vs. Decision Feedback, AWGN

Finally, we compare numerical results from the ROVA-based decision-feedback scheme in Sec. 3.3 and the two-phase information-feedback scheme earlier in this section. For the AWGN channel with SNR 2 dB, Fig. 3.10 displays the throughputs obtained for the optimal m=5 blocklengths for both schemes. Recall that the decision-feedback scheme uses tail-biting convolutional codes with the TB ROVA to determine when to stop, whereas the information-feedback scheme uses tail-biting convolutional codes in the communication phase and repetition codes in the confirmation phase to confirm or deny the receiver's tentative decisions.

At the shortest blocklength ($k = 16$), the two schemes perform similarly, but the decision-feedback scheme plateaus earlier than the information-feedback scheme, for both the 64-state codes and 1024-state codes. Even though two-phase scheme incurs a latency penalty of N_{conf} confirmation symbols during each round, this additional information allows the receiver to terminate earlier overall.

A key difference from a practical perspective is that the decision-feedback scheme requires only 1 bit of (noiseless) feedback per decoding attempt, whereas the information-feedback approach requires noiseless feedback of all I_i received symbols in the ith communication-phase transmission (or, equivalently, feedback of some other representation of the decoded message $\hat{W}_{N_i}$).

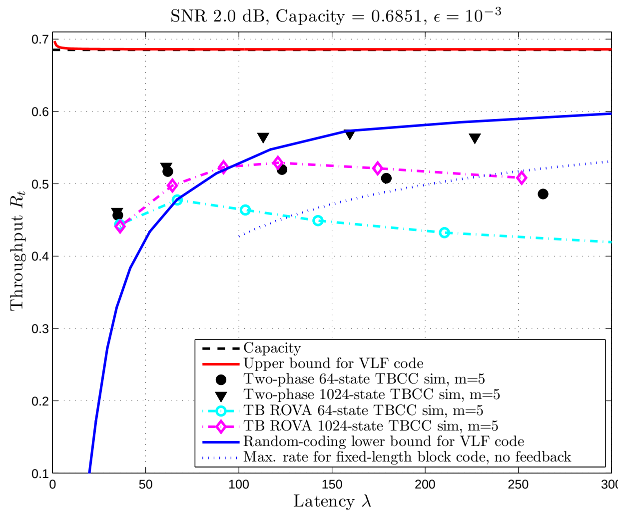


Figure 3.10: A comparison of the two-phase (information feedback) and ROVA-based approaches (decision feedback) for VLF codes on the AWGN channel. The two-phase scheme and the ROVA-based approach both use tail-biting convolutional codes (TBCCs) with $m = 5$ transmissions. In both cases, the undetected-error probability is required to be less than $\epsilon = 10^{-3}$.

3.4.7 Active Sequential Hypothesis Testing

This section compares the short-blocklength performance of the active sequential hypothesis testing scheme from Naghshvar et al. [2, 3] with the random-coding lower bound from Thm. 2. We begin by briefly reviewing the scheme presented in [2, 3], which is a deterministic, sequential coding scheme for DMCs using information feedback that achieves the optimal error-exponent. We then demonstrate via simulation that this scheme delivers rates significantly greater than those predicted by the random-coding lower bound.

After receiving each channel output symbol Y_n, the receiver computes the posterior probability of each message $i \in \mathcal{W}$, denoted as $\rho_i(n) = \mathrm{P}[W = i|Y^n]$. The belief (i.e., the posterior probability) of each message is updated according to:

$$\rho_i(n) = \frac{\rho_i(n-1)\mathrm{P}(Y=y|X=x_i)}{\sum\limits_{j=1}^{M} \rho_j(n-1)\mathrm{P}(Y=y|X=x_j)}, \tag{3.80}$$

where $x_i = f(W=i, Y^{n-1}=y^{n-1})$ is the encoder output at time n corresponding to message i and is based on the previous received symbols y^{n-1}. In this section, we assume equiprobable initial beliefs, i.e., $\rho_i(0) = \frac{1}{M} \forall\ i \in \mathcal{W}$, although any starting distribution is valid.

The encoder also computes the posterior probabilities $\rho_i(n)$, using the noiseless feedback of Y_n. After transmitting each symbol, the encoder partitions the M messages into two sets, S_1 and S_0, such that the sum of the posterior probabilities of the messages in each set is as close to one half as possible, i.e., $\sum\limits_{i\in S_0} \rho_i(n) \approx \sum\limits_{i\in S_1} \rho_i(n) \approx \frac{1}{2}$. For messages $i \in S_0$, the coded symbol sent is $x_i = 0$, and for messages $i \in S_1$, $x_i = 1$.

The stopping time τ is defined as

$$\tau = \min\{n \geq 0 : \max_{i\in\mathcal{W}} \rho_i(n) \geq 1-\epsilon\}, \tag{3.81}$$

which is the same stopping rule that we use for the ROVA in Sec. 3.3.1 with convolutional codes. Similar to the ROVA-based scheme, the error probability of this scheme is no more than $\epsilon > 0$, which is fixed. As noted in [3], this is a "possibly suboptimal stopping rule" in the sense that it is not guaranteed to yield the minimal average blocklength. The decoder's decision at time τ is $\hat{W}_\tau = \arg\max\limits_{i\in\mathcal{W}} \rho_i(\tau)$.

Three important quantities for the hypothesis testing scheme are the channel capacity C, the maximal relative entropy C_1, and C_2, the maximum likelihood ratio, defined as

$$C_2 = \max_{y\in\mathcal{Y}} \frac{\max\limits_{x\in\mathcal{X}} \mathrm{P}(Y|X=x)}{\min\limits_{x\in\mathcal{X}} \mathrm{P}(Y|X=x)}. \tag{3.82}$$

We assume that $C \leq C_1 \leq C_2 < \infty$.

For $M > 2$, the expected stopping time (average blocklength) is upper bounded as follows [2,3]:

$$\mathrm{E}[\tau] \leq \frac{\log\frac{1-\epsilon}{\epsilon}}{C_1} + \frac{\log(M-1)}{C} + \frac{3C_2^2}{CC_1}. \tag{3.83}$$

The bound above can be interpreted to mean that, on average, it takes roughly $\frac{\log(M-1)}{C}$ channel uses for the posterior $\rho_i(n)$ to go from $\frac{1}{M}$ to $\frac{1}{2}$, and another $\frac{\log\frac{1-\epsilon}{\epsilon}}{C_1}$ channel uses for $\rho_i(n)$ to reach $1-\epsilon$.

This bound is similar to an earlier result for symmetric DMCs, due to Burnashev [46, Thm. 2]:

$$\mathrm{E}[\tau] \leq \frac{\log\frac{1}{\epsilon}}{C_1} + \frac{\log M}{C} + C_{\mathcal{P}}, \tag{3.84}$$

where $C_{\mathcal{P}}$ is a constant determined by the transition probability matrix of the channel, $\mathcal{P}$. Burnashev's result is also based on an active sequential hypothesis testing (i.e., using information feedback).

Naghshvar et al. [2, 3] have used (3.83) to show that this deterministic coding scheme achieves the optimal error-exponent (by taking the limit as $\epsilon \to 0$ and allowing $M \to \infty$). As $\epsilon \to 0$ and $M \to \infty$, the constant term $\frac{3C_2^2}{CC_1}$ becomes negligible, but it is not obvious what impact this constant penalty has on short-blocklength performance. In fact, for the BSC($p = 0.05$) with $\epsilon = 10^{-3}$, the lower bound on rate corresponding to (3.83) is even looser than the random-coding lower bound of Thm. 2 (Fig. 3.11). Despite an elegant scheme that uses information feedback to carefully select each coded symbol X_n, the bound does not guarantee rates any better than what can be achieved with random coding and decision feedback.

However, it is not clear whether the coding scheme described in [2,3] can perform better than the bound given by (3.83). Note that (3.83) can be approximated as follows by ignoring the constant terms:

$$\mathrm{E}[\tau] \approx \frac{\log\frac{1-\epsilon}{\epsilon}}{C_1} + \frac{\log(M-1)}{C}. \tag{3.85}$$

A Monte Carlo simulation of the hypothesis testing scheme, based on Algorithm 2 in [3] and shown in Fig. 3.11, demonstrates that the approximation of (3.85) closely approximates the simulated short-blocklength performance. This active sequential hypothesis test-

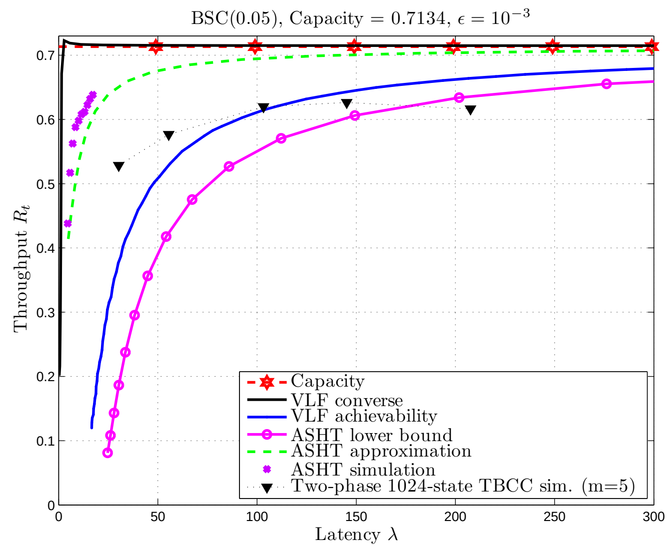


Figure 3.11: Monte Carlo simulations of active sequential hypothesis testing (ASHT) from Naghshvar et al. [2, 3] demonstrate the benefits of information feedback compared to decision-feedback VLF achievability based on random coding. Also shown are results from the m=5 two-phase incremental redundancy scheme from Sec. 3.4.1.

ing (ASHT) simulation delivered rates that surpassed both the lower bound in (3.83) and the random-coding lower bound of Thm. 2 (Fig. 3.11).

Algorithm 2 in [3] has complexity of order M^2 (at the transmitter), so it is computationally challenging to evaluate the ASHT scheme for larger message sizes. (For comparison, ML decoding in general has complexity of order M.) Fig. 3.11 uses $k = \log_2 M \in \{2, 3, \dots, 11\}$. At least 25 message errors were obtained for each value of k in the simulation.

Naghshvar et al.'s ASHT scheme [3] assumes that feedback is sent after every received symbol Y_{n-1}, and uses the feedback to determine each of the subsequent transmitted symbols X_n. In contrast, the m=5 two-phase incremental redundancy scheme from this chapter uses packet transmissions and only requires feedback to be sent after each packet. Feedback from the communication-phase packet is used to determine the repetition-coded ACK or NACK packet in the confirmation phase, rather than to determine individual coded symbols. In this sense, the ASHT scheme is more "active" because it responds to feedback at the smallest possible interval (after every received symbol). The performance of the two-phase scheme for the BSC is shown in Fig. 3.11. The ASHT scheme approaches capacity much faster, reaching 89.5% of capacity at an average blocklength 17.23 bits (k=11). However, computational complexity has prevented us from simulating higher values of k (and higher latencies) for the ASHT scheme. The packet-based nature of the two-phase scheme has significantly lower computational requirements that make it more attractive for practical implementation.

3.5 Conclusion

This chapter demonstrated a reliability-based decision-feedback scheme that provides throughput surpassing the random-coding lower bound at short blocklengths. We selected convolutional codes for their excellent performance at short blocklengths and used the tail-biting ROVA to avoid the rate loss of terminated convolutional codes. For both the BSC and AWGN channels, convolutional codes provided throughput above 77% of capacity, with blocklengths less than 150 bits. While codes with higher constraint lengths would be expected to provide superior performance, computational considerations limited us to evaluate 64-state codes with decoding after every symbol and 1024-state codes in an m=5 transmission incremental redundancy setting. We introduced a novel blocklength-selection algorithm to aid in selecting the m=5 transmission lengths and showed that despite the limitations on decoding frequency, the incremental redundancy scheme is competitive with decoding after every symbol. Finally, we demonstrated that the latency overhead of CRCs imposes a severe rate-loss penalty at short blocklengths, whereas reliability-based decoding does not require transmission of separate error-detection bits.

This chapter also demonstrated two information-feedback coding schemes, namely two-phase incremental redundancy and active sequential hypothesis testing. Similar to the m=5 ROVA-based incremental redundancy scheme, we showed how the blocklength-selection algorithm could be used to optimize the throughput for the two-phase scheme. The combination of tail-biting convolutional codes and repetition codes in the two-phase scheme delivered higher rates than the tail-biting convolutional codes in the decision-feedback setting. This improvement is due to the adaptive nature of information-feedback codes. Similarly, the active sequential hypothesis testing scheme provided even higher rates than the two-phase scheme, in exchange for more frequent decoding attempts and feedback opportunities.

Most importantly, this chapter has shown that feedback codes, due to their variable-length structure, offer tremendous gains in rate at short blocklengths compared to fixed-length codes without feedback. The various assumptions in this chapter on the amount of noiseless feedback and on the decoding intervals must be closely examined for practical systems to determine what is realistic. It is expected that the m-transmission decision-feedback setting, requiring only a single bit of feedback per packet transmission, is the most practically relevant in general. However, systems operating in the short-blocklength regime that can accommodate some degree of "active" encoding and more-frequent decoding, as in the two-phase scheme, will benefit from higher rates than can be achieved with decision-feedback alone.

3.6 Appendix

3.6.1 Numerical Computation of the VLF Lower Bound

For channels with bounded information density, Wald's equality (also known as Wald's identity or Wald's lemma) allows us to compute an upper bound on the expected stopping time $\mathrm{E}[\tau]$ in the random-coding lower bound of Thm. 2 as follows:

$$\mathrm{E}[\tau] \leq \frac{\log(M-1) + \log\frac{1}{\epsilon} + B}{C}, \tag{3.86}$$

where $B < \infty$ is the upper bound on the information density:

$$B = \sup_{x\in\mathcal{X},\ y\in\mathcal{Y}} i(x;y). \tag{3.87}$$

Proof. Defining $S_j = i(X_j; Y_j) = \log \frac{P(Y_j|X_j)}{P(Y_j)}$, we have

$$
\begin{aligned}
S^n = i(X^n; Y^n) &= \log \frac{P(Y^n|X^n)}{P(Y^n)} && (3.88)\\
&= \sum_{j=1}^{n} \log \frac{P(Y_j|X_j)}{P(Y_j)} && (3.89)\\
&= \sum_{j=1}^{n} S_j, && (3.90)
\end{aligned}
$$

where (3.89) follows due to the random codebook generation. Since the S_j are i.i.d with $\mathrm{E}[S_j] = C$, Wald's equality gives the following result [69, Ch. 5] :

$$
\mathrm{E}[S^\tau] = \mathrm{E}[\tau]\mathrm{E}[S_1]. \tag{3.91}
$$

This leads to the following upper bound on $\mathrm{E}[\tau]$:

$$
\begin{aligned}
\mathrm{E}[\tau] &= \frac{\mathrm{E}[S^\tau]}{C} && (3.92)\\
&= \frac{\mathrm{E}[S^{\tau-1} + S_\tau]}{C} && (3.93)\\
&\leq \frac{\gamma + B}{C}, && (3.94)
\end{aligned}
$$

where (3.94) follows from the definition of the threshold γ in the random-coding lower bound and of B above.

Based on the fact that we can pick $\gamma = \log \frac{M-1}{\epsilon}$ in order to satisfy the ϵ error constraint, we have the following:

$$
\mathrm{E}[\tau] \leq \frac{\log \frac{M-1}{\epsilon} + B}{C}, \tag{3.95}
$$

which proves (3.86). □

Examples. For the BSC with crossover probability p, $B = \log 2(1-p)$. For the binary-input AWGN channel, $B = \log 2$ (regardless of the signal-to-noise ratio). □

3.6.2 Maximal Relative Entropy

A number of the bounds in Polyanskiy et al. [25] and Naghshvar et al. [2, 3] are limited to channels with finite maximal relative entropy. In this section, we review the definition of the maximal relative entropy and derive this quantity for the binary symmetric channel and the binary-input AWGN channel with soft-decision decoding.

The maximal relative entropy C_1 is defined as

$$C_1 = \max_{x_1,x_2\in\mathcal{X}} D\left(\mathrm{P}(Y|X=x_1)||\mathrm{P}(Y|X=x_2)\right) \tag{3.96}$$

$$= \max_{x_1,x_2\in\mathcal{X}} \mathrm{E}_{\mathrm{P}(Y|X=x_1)}\left[\log\frac{\mathrm{P}(Y|X=x_1)}{\mathrm{P}(Y|X=x_2)}\right]. \tag{3.97}$$

Examples. Due to the symmetry of the binary symmetric channel with crossover probability p, we can write the maximal relative entropy C_1^{BSC} as

$$C_1^{\mathrm{BSC}} = \mathrm{E}_{\mathrm{P}(Y|X=0)}\left[\log\frac{\mathrm{P}(Y|X=0)}{\mathrm{P}(Y|X=1)}\right] \tag{3.98}$$

$$= \mathrm{E}_{\mathrm{P}(Y|X=0)}\left[1\{Y=0\}\log\frac{1-p}{p} + 1\{Y=1\}\log\frac{p}{1-p}\right] \tag{3.99}$$

$$= (1-p)\log\frac{1-p}{p} + p\log\frac{p}{1-p} \tag{3.100}$$

$$= (1-2p)\log\frac{1-p}{p}. \tag{3.101}$$

For the binary-input AWGN with soft-decision decoding, when the SNR is $\eta = \frac{P}{\sigma^2}$, the channel inputs are $x \in \{+\sqrt{P}, -\sqrt{P}\}$ and the noise variance is σ^2. Without loss of generality, we can pick $x_1 = +\sqrt{P}$ and $x_2 = -\sqrt{P}$, so that the maximal relative entropy $C_1^{\text{BI-AWGN}}$ is

$$C_1^{\text{BI-AWGN}} = \mathrm{E}_{\mathrm{P}(Y|X=+\sqrt{P})}\left[\log\frac{\mathrm{P}(Y|X=+\sqrt{P})}{\mathrm{P}(Y|X=-\sqrt{P})}\right] \tag{3.102}$$

$$= \mathrm{E}_{\mathrm{P}(Y|X=+\sqrt{P})}\left[\log\frac{\exp\{-\frac{1}{2\sigma^2}(Y-\sqrt{P})^2\}}{\exp\{-\frac{1}{2\sigma^2}(Y+\sqrt{P})^2\}}\right] \tag{3.103}$$

$$= \mathrm{E}_{\mathrm{P}(Y|X=+\sqrt{P})}\left[\frac{2\sqrt{P}Y}{\sigma^2}\right] \tag{3.104}$$

$$= \frac{2\sqrt{P}}{\sigma^2}\ \mathrm{E}_{\mathrm{P}(Y|X=+\sqrt{P})}[Y] \tag{3.105}$$

$$= \frac{2P}{\sigma^2} \tag{3.106}$$

$$= 2\eta, \tag{3.107}$$

where (3.106) follows because the mean of Y under $\mathrm{P}(Y|X=+\sqrt{P})$ is $\sqrt{P}$.

Thus, we have showed for both the BSC and the BI-AWGN channel that $C_1 < \infty$. $\square$

3.6.3 Proof of Cor. 1

Proof of Random-coding lower bound for repeat-after-N codes. The proof closely follows that of [25, Theorem 3]. We define M stopping times τ_j for the jth codeword:

$$\tau_j = \inf\{n \geq 0 : i_N(X^n(j); Y^n) \geq \gamma\}, \tag{3.108}$$

where $X^n(j)$ is the first n symbols of the jth codeword. At each n, the decoder evaluates the M information densities $i_N(X^n(j); Y^n)$ and makes a final decision at time τ^* when the first of these (possibly more than one at once) reaches the threshold γ:

$$\tau^* = \min_{j=1,\ldots,M} \tau_j. \tag{3.109}$$

The decoder at time τ^* selects codeword $m = \max\{j : \tau_j = \tau^*\}$.

The average blocklength $\ell = \mathrm{E}[\tau^*]$ is upper bounded as follows:

$$\mathrm{E}[\tau^*] \leq \frac{1}{M}\sum_{j=1}^{M} \mathrm{E}[\tau_j | W = j] \tag{3.110}$$

$$= \mathrm{E}[\tau_1 | W = 1] \tag{3.111}$$

$$= \mathrm{E}[\tau] \tag{3.112}$$

$$= \sum_{n=0}^{\infty} \mathrm{P}[\tau > n] \tag{3.113}$$

$$= \left(1 + \mathrm{P}[\tau > N] + \mathrm{P}[\tau > N]^2 + \dots\right) \sum_{n=0}^{N-1} \mathrm{P}[\tau > n] \tag{3.114}$$

$$= \frac{\sum_{n=0}^{N-1} \mathrm{P}[\tau > n]}{1 - \mathrm{P}[\tau > N]}. \tag{3.115}$$

Eq. (3.111) follows from the symmetry of the M stopping times and (3.112) is by the definition of τ as given in (3.19). Because the modified information densities depend only on the symbols in the current N-block, repeat-after-N VLF codes satisfy the following property, which leads to (3.114):

$$\mathrm{P}[\tau > n] = \mathrm{P}[\tau > N]^s \mathrm{P}[\tau > r], \tag{3.116}$$

where $n = sN + r$. The condition that $\mathrm{P}[\tau \leq N] > 0$ in Cor. 1 is required so that $\mathrm{P}[\tau > N] < 1$, guaranteeing that the sum in (3.114) will converge.

Using $\hat{W}_n$ to denote the decoder's decision at time n, an error occurs if the decoder chooses $\hat{W}_{\tau^*} \neq W$. The probability of error ϵ can be bounded due to the random codebook generation:

$$\epsilon = \mathrm{P}[\hat{W}_{\tau^*} \neq W] \tag{3.117}$$
$$\leq \mathrm{P}[\hat{W}_{\tau^*} \neq 1|W=1] \tag{3.118}$$
$$\leq \mathrm{P}[\tau_1 \geq \tau^*|W=1] \tag{3.119}$$
$$\leq \mathrm{P}\Big[\bigcup_{j=2}^{M}\{\tau_1 \geq \tau_j\}|W=1\Big] \tag{3.120}$$
$$\leq (M-1)\mathrm{P}[\tau_1 \geq \tau_2|W=1] \tag{3.121}$$
$$= (M-1)\mathrm{P}[\tau \geq \bar{\tau}]. \tag{3.122}$$

The last line follows because $(\tau, \bar{\tau})$ have the same distribution as (τ_1, τ_2) conditioned on $W = 1$. □

3.6.4 Proof of Thm. 3

Proof of two-phase, information-feedback lower bound (fixed-length confirmation phase). Due to the random-coding construction as in Cor. 1, the expected length of the first phase is upper bounded by $\mathrm{E}[\tau]$. The latency in the second phase is always N_{conf}. Because the transmission restarts when a NACK is received, we divide the overall latency by the complement of the probability of receiving a NACK at the receiver, as in the proof of Cor. 1. Thus the average blocklength ℓ is upper bounded according to

$$\ell \leq \frac{\mathrm{E}[\tau] + N_{\text{conf}}}{1 - \mathrm{P}(\mathcal{N})}, \tag{3.123}$$

proving (3.29). The condition that $\mathrm{P}(\mathcal{N}) < 1$ in Thm. 3 is required so the geometric series implied by (3.29) will converge.

Using $\hat{W}_n$ to denote the decoder's decision at time n, an error occurs if the decoder chooses $\hat{W}_{\tau^*} \neq W$ (a communication-phase error) and if a NACK is decoded as an ACK (a confirmation-phase error), or if this combination occurs after the receiver has decoded one or more consecutive NACKs. The probability of error ϵ is then given by

$$\epsilon = \frac{\mathrm{P}[\hat{W}_{\tau^*} \neq W] p_{n \to a}}{1 - \mathrm{P}(\mathcal{N})}. \tag{3.124}$$

As in the proof of Cor. 1, the communication-phase error can be bounded due to the random codebook generation:

$$\mathrm{P}[\hat{W}_{\tau^*} \neq W] \leq (M-1)\mathrm{P}[\bar{\tau} \leq \tau]. \tag{3.125}$$

Combining (3.124) and (3.125), we have

$$\epsilon \leq \frac{(M-1)\mathrm{P}[\bar{\tau} \leq \tau] p_{n \to a}}{1 - \mathrm{P}(\mathcal{N})}, \tag{3.126}$$

proving (3.30). □

3.6.5 Numerical Computation of Thm. 3

This section describes how to numerically evaluate the information-feedback lower bound (with a fixed-length confirmation phase) of Thm. 3. In order to plot the best possible rate for this information-feedback lower bound, we need to optimize over the threshold γ and the confirmation blocklength N_{conf}, given a fixed message cardinality M and target error ϵ'. The decision-feedback VLF lower bound from Thm. 2, plotted in earlier sections, is somewhat more straightforward, since only γ needs to be chosen. The two-parameter optimization in this bound is more difficult.

We begin by considering the following bound on $\mathrm{P}[\bar{\tau} \leq n]$:

$$\begin{aligned}
\mathrm{P}[\bar{\tau} \leq n] &= \mathrm{E}[1\{\bar{\tau} \leq n\}] && (3.127)\\
&= \mathrm{E}[1\{\tau \leq n\} \exp\{-i(X^\tau; Y^\tau\}] && (3.128)\\
&\leq \exp\{-\gamma\}, && (3.129)
\end{aligned}$$

where the last inequality follows because $i(X^\tau; Y^\tau) \geq \gamma$ by definition in (3.31). Accordingly, we can write the following

$$\mathrm{P}[\bar{\tau} \leq \tau] = \sum_{n=0}^{\infty} \mathrm{P}[\tau = n]\mathrm{P}[\bar{\tau} \leq n] \tag{3.130}$$

$$\leq \sum_{n=0}^{\infty} \mathrm{P}[\tau = n] \exp\{-\gamma\} \tag{3.131}$$

$$= \exp\{-\gamma\}, \tag{3.132}$$

which leads to a looser bound on the average probability of error as follows:

$$\epsilon \leq \frac{(M-1)\exp\{-\gamma\}p_{n\to a}}{1-\mathrm{P}(\mathcal{N})}. \tag{3.133}$$

The average communication-phase blocklength $\mathrm{E}[\tau]$ can be computed using Wald's equality, as described in [77], or it may be computed as follows:

$$\mathrm{E}[\tau] = \sum_{n=0}^{\infty} \mathrm{P}[\tau > n] \tag{3.134}$$

$$\leq \sum_{n=0}^{\infty} \mathrm{P}[i(X^n;Y^n) < \gamma]. \tag{3.135}$$

For each value of n, the term $\mathrm{P}[i(X^n;Y^n) < \gamma]$ can be evaluated numerically. For the AWGN channel, this computation involves a 3-dimensional integral for each n. The BSC computation is simpler: $\mathrm{P}[i(X^n;Y^n) < \gamma]$ can be rearranged in terms of the C.D.F. of a binomial random variable.

For a given error constraint ϵ', we pick a starting confirmation block $N_0 = 1$ and then find γ_0, the smallest γ such that the right-hand side of (3.133) is smaller than ϵ' (which guarantees that $\epsilon < \epsilon'$). We then evaluate the right-hand side of (3.29) to find the average blocklength ℓ corresponding to these parameters. We search for the best parameters by increasing γ_0 and increasing N_0 until the optimum is found.

The rate for each (M, ϵ) pair may be further improved (with additional computational complexity) by considering a skewed hypothesis test for deciding ACK or NACK at the receiver, based on the N_{conf} confirmation symbols. The balanced hypothesis test would have

$p_{n\to a} = p_{a\to n}$, but this is not necessarily optimal. Because decoding a NACK results in a costly restart, it may be preferable to skew the receiver's hypothesis test towards ACK, since errors in both the communication phase and confirmation phase should be rare.

As an example, consider the following ACK/NACK decision rule for the BSC with threshold N_{th} at the receiver ($0 \leq N_{\text{th}} \leq N_{\text{conf}}$):

- $\mathcal{H}_0$: Decide ACK if at least N_{th} of N_{conf} bits are 1's.
- $\mathcal{H}_1$: Decide NACK if less than N_{th} of N_{conf} bits are 1's.

Introducing the threshold N_{th} complicates the lower bound computation because N_{th} must be optimized for each value of N_{conf}. Some time can be saved by recognizing that since NACKs are costly in terms of latency, $N_{\text{t}} > \lceil \frac{N}{2} \rceil$ will not be selected and need not be evaluated. We repeated this optimization process across a wide range of M values (and fixed ϵ), ultimately arriving at the curve in Fig. 3.2, which includes an optimization over the skew threshold N_{th}.

3.6.6 Numerical Computation of Thm. 4

This section describes how to numerically evaluate the information-feedback lower bound (with a variable-length confirmation phase) of Thm. 4. Note that the expected stopping time of the confirmation phase, $\mathrm{E}[\mu]$, can be upper bounded as follows [71]:

$$\mathrm{E}[\mu|\mathcal{H}_0] \leq \frac{1}{D_{c,e}} \left[p_{a\to a}(-t_0 + B) - p_{a\to n}\ t_1\right] \tag{3.136}$$

$$\mathrm{E}[\mu|\mathcal{H}_1] \leq \frac{1}{D_{e,c}} \left[p_{n\to n}(t_1 + B) + p_{n\to a}\ t_0\right], \tag{3.137}$$

where

$$D_{c,e} = D\big(P(Y_i|X_i = x_{\mathrm{c}})||P(Y_i|X_i = x_{\mathrm{e}})\big) \tag{3.138}$$

$$= -\mathrm{E}_{\mathcal{H}_0}[\Lambda_i(Y)], \tag{3.139}$$

$$D_{e,c} = D\big(P(Y_i|X_i = x_{\mathrm{e}})||P(Y_i|X_i = x_{\mathrm{c}})\big) \tag{3.140}$$

$$= \mathrm{E}_{\mathcal{H}_1}[\Lambda_i(Y)], \tag{3.141}$$

$$B = \sup|\Lambda_i(Y)|. \tag{3.142}$$

That is, B is the upper bound on $\Lambda_i(Y)$ under either hypothesis. Putting together (3.136) and (3.137), we have

$$\mathrm{E}[\mu] = \mathrm{E}[\mu|\mathcal{H}_0]\mathrm{P}[\mathcal{H}_0] + \mathrm{E}[\mu|\mathcal{H}_1]\mathrm{P}[\mathcal{H}_1] \tag{3.143}$$

$$\leq \frac{1}{D_{c,e}}\big(p_{a\to a}(-t_0 + B) - p_{a\to n}\ t_1\big) + \frac{(M-1)\mathrm{P}[\bar{\tau} \leq \tau]}{D_{e,c}}\big(p_{n\to n}(t_1 + B) + p_{n\to a}\ t_0\big) \tag{3.144}$$

$$\leq \frac{1}{D_{c,e}}\big(-t_0 + B\big) + \frac{(M-1)\mathrm{P}[\bar{\tau} \leq \tau]}{D_{e,c}}\big(t_1 + B\big) \tag{3.145}$$

Because the exponential error bounds from Wald's SPRT [70] in (3.37) and (3.36) are upper bounds on $p_{a\to n}$ and $p_{n\to a}$, they cannot be applied to all of the confirmation-phase error terms in (3.144), leading to the looser bound on $\mathrm{E}[\mu]$ in (3.145).

Example. For the binary symmetric channel with crossover probability p, the bound on the log likelihood-ratios is $B = \log\frac{1-p}{p}$, and the average value is

$$\mathrm{E}_{\mathcal{H}_0}[\Lambda_1(Y)] = (1-2p)\log\frac{1-p}{p} = C_1. \tag{3.146}$$

Since the BSC is symmetric, $D_{c,e} = D_{e,c} = C_1$, which allows us to particularize (3.145) as

$$\mathrm{E}[\mu] \leq \frac{1}{C_1}\left[-t_0 + B + (M-1)\exp\{-\gamma\}\big(t_1 + B\big)\right]. \tag{3.147}$$

□

However, because the bound in (3.145) is not always tight, a better approach may be to evaluate $\mathrm{E}[\mu]$ in the following manner, similar to the computation of $\mathrm{E}[\tau]$ in Sec. 3.6.5.

$$\mathrm{E}[\mu] = \sum_{n=0}^{\infty} \mathrm{P}[\mu > n] \tag{3.148}$$

$$\leq \sum_{n=0}^{\infty} \mathrm{P}[t_0 < \Lambda^n < t_1], \tag{3.149}$$

since $\mathrm{P}[\mu > n] \leq \mathrm{P}[t_0 < \Lambda^n < t_1]$ by the definition of μ. For each value of n, the term $\mathrm{P}[t_0 < \Lambda^n < t_1]$ can be evaluated numerically. As we will see for the BSC, this probability can be rearranged in terms of C.D.F.s of a binomial random variable.

Example. For the BSC(p) with $p < \frac{1}{2}$, we assume w.l.o.g. that the transmitter sends x_c=0 under $\mathcal{H}_0$ and x_e=1 under $\mathcal{H}_1$ in the confirmation phase. Then the number of ones in the received sequence Y^n is a random variable d, distributed $\sim$Bin(n, p_i), where $p_0 = p$ under $\mathcal{H}_0$ and $p_1 = 1 - p$ under $\mathcal{H}_1$.

The log likelihood-ratio is

$$\Lambda^n = \log \frac{P(Y^n|X^n = x_\mathrm{e}^n)}{P(Y^n|X^n = x_\mathrm{c}^n)} \tag{3.150}$$

$$= \log \frac{p^{n-d}(1-p)^d}{(1-p)^{n-d}p^d} \tag{3.151}$$

$$= 2d \log \frac{1-p}{p} - n \log \frac{1-p}{p} \tag{3.152}$$

$$= 2dB - nB, \tag{3.153}$$

where $B = \log \frac{1-p}{p} > 0$.

The upper bound on the retransmission probability is

$$\mathrm{P}[\mu > n] \leq \mathrm{P}[t_0 < \Lambda^n < t_1] \tag{3.154}$$

$$= \mathrm{P}[t_0 < 2dB - nB < t_1] \tag{3.155}$$

$$= \mathrm{P}[nB + t_0 < 2dB < nB + t_1] \tag{3.156}$$

$$= \mathrm{P}\left[\frac{nB + t_0}{2B} < d < \frac{nB + t_1}{2B}\right], \tag{3.157}$$

where (3.157) can be computed from the C.D.F. of $d \sim$Bin(n, p_i).

Thus, we can separately compute $\mathrm{E}[\mu|\mathcal{H}_0]$ and $\mathrm{E}[\mu|\mathcal{H}_1]$ to yield $\mathrm{E}[\mu]$ as follows:

$$\mathrm{E}[\mu|\mathcal{H}_0] \leq \sum_{n=0}^{\infty} \mathrm{P}\left[\frac{nB + t_0}{2B} < d < \frac{nB + t_1}{2B}\right],\ d \sim \mathrm{Bin}(n, p_0) \tag{3.158}$$

$$\mathrm{E}[\mu|\mathcal{H}_1] \leq \sum_{n=0}^{\infty} \mathrm{P}\left[\frac{nB + t_0}{2B} < d < \frac{nB + t_1}{2B}\right],\ d \sim \mathrm{Bin}(n, p_1) \tag{3.159}$$

$$\mathrm{E}[\mu] = \mathrm{E}[\mu|\mathcal{H}_0]\mathrm{P}[\mathcal{H}_0] + \mathrm{E}[\mu|\mathcal{H}_1]\mathrm{P}[\mathcal{H}_1]. \tag{3.160}$$

We used the above approach to illustrate the achievability in Thm. 4 for the BSC(p=0.05) with $\epsilon = 10^{-3}$ in Fig. 3.2, rather than the looser bound on $\mathrm{E}[\mu]$ in (3.145).

Similar to the blocklength-selection algorithm of Sec. 3.6.7, we used a coordinate-descent algorithm to minimize $\mathrm{E}[\mu]$ over the parameters γ, t_1, and t_0, subject to the error constraint ϵ in (3.42). Recognizing that this two-phase bound can only improve upon the one-phase decision-feedback bound in Thm. 2 if the communication phase is shorter, the search for the optimal γ can be limited to the range $[\gamma_{\min}, \gamma_{\max}]$, where $\gamma_{\min}$ is the smallest γ that meets the error constraint (given t_1 and t_0) and $\gamma_{\max}$ is the threshold used for the decision-feedback bound. □

3.6.7 General Blocklength-selection Algorithm

Selecting the m incremental transmission lengths $\{I_i\}$ that minimize the latency (or equivalently, maximize the throughput) of VLF coding schemes is non-trivial. The complexity of a brute-force search grows exponentially with m for a fixed k, and grows linearly with k for a fixed m. In this section, we describe an efficient blocklength-selection algorithm that can be used to identify suitable blocklengths for general incremental redundancy schemes. The goal of the algorithm is to select the m integer-valued incremental transmission lengths $\{I_i\}$ (and possibly the integer-valued confirmation blocklength N_{conf}, if using a two-phase scheme) as follows:

$$\{I_i\}^*, N^*_{\text{conf}} = \arg \min_{\{I_i\}, N_{\text{conf}} \in \mathbb{Z}} \lambda \ \text{ s.t. } \ P(\text{UE}) \leq \epsilon. \tag{3.161}$$

For the decision-feedback scheme using the TB ROVA in Sec. 3.3, the probability of undetected error is less than ϵ by definition, so the constraint can be ignored. Note the objective function λ in (3.161) is in general not convex. Even in the simple case when the retransmission probabilities are survivor functions of a chi-square distribution as in [55], when $\mathrm{P}_{\text{re}}(N) = 1 - F_{\chi^2_N}(r^2_N)$, λ is not convex. The latency λ for the two-phase approach in Sec. 3.4.3 is even more complicated, since the probability of retransmission (i.e., the probability of decoding a NACK at the receiver) that is approximated by (3.66) has several terms, and because the confirmation blocklength N_{conf} must also be optimized.

Algorithm 1 illustrates the proposed blocklength-selection algorithm for an m-transmission incremental redundancy scheme without confirmation blocks. (Sec. 3.6.9 describes how the algorithm can be applied to the two-phase scheme with confirmation blocks.) Starting from a pseudo-random initial vector $\{I_1, I_2, \ldots, I_m\}$, the algorithm employs performs coordinate descent, wherein one transmission length I_i is optimized while all others are held fixed. (See function single_coord_search( ) in Algorithm 2.) Via the function compute_objective( ), the objective function λ is evaluated for positive and negative unit steps in increment I_i and I_i is updated if the objective improves (Steps (9) and (11) in Algorithm 2).

Once the objective cannot be improved by any single-coordinate steps, diagonal steps from each of the possible two-coordinate pairs (I_i, I_j) are evaluated. (See function double_coord_search( ) in Algorithm 3.) The function $pair(n)$ returns the coordinate indices $(i, j) \in \{1, \ldots, m\} \times \{1, \ldots, m\}$ that represent one of these $\binom{m}{2}$ pairs. For each two-coordinate pair, four possible neighboring diagonal steps are evaluated. The transmission lengths I_i and I_j are updated if the best of the four diagonal steps improves the objective λ. This continues until the objective cannot be improved by additional diagonal steps.

The entire process then starts over from another pseudo-random initial vector. Random restarts are employed in order to avoid getting stuck at local optima of λ, of which there can be many.

Empirical trials of this algorithm for several different families of retransmission probabilities demonstrated significantly reduced computation time compared to a brute-force approach (which is in general not possible for large k). Furthermore, while this algorithm is not guaranteed to find the global optimum, results show that the final objective value λ_{best} was improved compared to results from an earlier quasi-brute-force trial.

Algorithm 1: General algorithm for selecting optimal transmission lengths $\{I_i\}$.

for $r = 1$ ***to*** *max_restarts* **do**
- Pick pseudo-random initial lengths $\{I_1, \ldots, I_m\}$;
- Re-order initial lengths so that I_1 is largest;
- $\lambda \leftarrow 10^6$, $\lambda_{\text{best}} \leftarrow 10^6$, $resolution \leftarrow 10^{-3}$;
- $(\{I_i\}, \lambda) \leftarrow single\_coord\_search(\{I_i\}, \lambda, \lambda_{\text{best}}, resolution)$ (Alg. 2);
- $(\{I_i\}, \lambda) \leftarrow double\_coord\_search(\{I_i\}, \lambda, \lambda_{\text{best}}, resolution)$ (Alg. 3);

Algorithm 2: single_coord_search($\{I_i\}, \lambda, \lambda_{\text{best}}, resolution$): Single-coordinate descent portion of Algorithm 1.

---

for $t = 1$ ***to*** $max\_iterations$ **do**
    $\lambda_{\text{last}} \leftarrow \lambda$;
    for $i = 1$ ***to*** m **do**
        $break\_flag \leftarrow 0$;
        while $break\_flag == 0$ **do**
            $\lambda \leftarrow compute\_objective(I_1, \ldots, I_i, \ldots, I_m)$;
            $\lambda^+ \leftarrow compute\_objective(I_1, \ldots, I_i + 1, \ldots, I_m)$;
            $\lambda^- \leftarrow compute\_objective(I_1, \ldots, I_i - 1, \ldots, I_m)$;
            if $\lambda^+ \leq \lambda$ **then**
                $I_i \leftarrow I_i + 1$, continue;
            if $\lambda^- < \lambda$ **then**
                $I_i \leftarrow I_i - 1$, continue;
            $break\_flag \leftarrow 1$;
    $\lambda \leftarrow compute\_objective(I_1, \ldots I_m)$;
    if $\lambda < \lambda_{best}$ **then**
        $\lambda_{\text{best}} \leftarrow \lambda$, $\{I_i\}_{\text{best}} \leftarrow \{I_i\}$;
    if $\lambda_{last} - \lambda < resolution$ **then**
        break;

Algorithm 3: double_coord_search($\{I_i\}, \lambda, \lambda_{\text{best}}, resolution$): Double-coordinate (diagonal) descent portion of Algorithm 1.

for $t = 1$ ***to*** $max\_iterations$ **do**
    $\lambda_{\text{last}} \leftarrow \lambda$;
    for $n = 1$ ***to*** $\binom{m}{2}$ **do**
        $(i, j) \leftarrow pair(n)$;
        $break\_flag \leftarrow 0$;
        while $break\_flag == 0$ **do**
            $\lambda \leftarrow compute\_objective(I_1, \ldots, I_i, \ldots, I_j, \ldots I_m)$;
            $\lambda^{++} \leftarrow compute\_objective(I_1, \ldots, I_i + 1, \ldots, I_j + 1, \ldots I_m)$;
            $\lambda^{+-} \leftarrow compute\_objective(I_1, \ldots, I_i + 1, \ldots, I_j - 1, \ldots I_m)$;
            $\lambda^{-+} \leftarrow compute\_objective(I_1, \ldots, I_i - 1, \ldots, I_j + 1, \ldots I_m)$;
            $\lambda^{--} \leftarrow compute\_objective(I_1, \ldots, I_i - 1, \ldots, I_j - 1, \ldots I_m)$;
            $\lambda_{\text{min}} \leftarrow min(\lambda^{++}, \lambda^{+-}, \lambda^{-+}, \lambda^{--})$;
            if $\lambda_{min} < \lambda$ **then**
                $I_i \leftarrow I_i(\lambda_{\text{min}})$, continue;
            $break\_flag \leftarrow 1$;
    $\lambda \leftarrow compute\_objective(I_1, \ldots I_m)$;
    if $\lambda < \lambda_{best}$ **then**
        $\lambda_{\text{best}} \leftarrow \lambda$, $\{I_i\}_{\text{best}} \leftarrow \{I_i\}$;
    if $\lambda_{last} - \lambda < resolution$ **then**
        break;

3.6.8 TB ROVA Version of Blocklength-selection Algorithm

For the decision-feedback scheme of Sec. 3.3 that uses the TB ROVA to compute the posterior probability of the decoded word, the probability of retransmission $\mathrm{P}_{\mathrm{re}}(N)$ is the probability that the posterior at blocklength N is less than $(1-\epsilon)$, which is difficult to determine analytically.[6] Instead, we obtained estimates of the empirical retransmission-probabilities of the rate-1/3 convolutional codes in Table 3.1 and used those estimates of $\mathrm{P}_{\mathrm{re}}(N)$ to compute the objective λ in the algorithm.

To do so, we first simulated fixed-length transmission of rate-1/3, tail-biting convolutional codes from Table 3.1 at a small number of pseudo-randomly punctured blocklengths N_{sim} for each fixed message-size k, where $N_{\mathrm{sim}} \in \{k : \frac{k}{4} : 3k\}$. For each (k, N_{sim}) pair, we counted the number of decoded words with posterior probability less than $(1-\epsilon)$, indicating that a retransmission would be required, until there were at least 100 codewords that would trigger a retransmission. We computed $\mathrm{P}_{\mathrm{re}}(N_{\mathrm{sim}})$ according to

$$\mathrm{P}_{\mathrm{re}}(N_{\mathrm{sim}}) = \frac{\#\text{ codewords triggering retransmission}}{\text{total }\#\text{ codewords simulated}}. \tag{3.162}$$

The full set of estimated retransmission probabilities $\tilde{\mathrm{P}}_{\mathrm{re}}(N)$ for $N \in \{1, \ldots, 3k\}$ was then obtained by a log-polynomial interpolation of the simulated values of $\mathrm{P}_{\mathrm{re}}(N_{\mathrm{sim}})$. Finally, the estimated probabilities were used in the algorithm to select the optimal transmission lengths $\{I_i\}^*$. The performance of the decision-feedback VLF scheme using these m=5 optimal blocklengths is evaluated in Sec. 3.3. We used $max\_restarts = 100$ in our implementation.

3.6.9 Two-phase Version of Blocklength-selection Algorithm

The two-phase scheme with confirmation blocks operates in the same manner as above, except that there are $(m+1)$ integer coordinates to optimize instead of m, and additional

[6] For a given convolutional code, the weight spectrum for each blocklength N could be used to bound or approximate the posterior probability and that could be used to bound or approximate the retransmission probability, but spectrum-based approaches tend not to be tight over a wide range of SNRs. Further complicating the task is the weight spectrum must be based on a rate-compatible, puncturing pattern. Instead of optimizing this puncturing pattern, we use the same pseudo-random puncturing pattern throughout.

conditions must be added to ensure that any steps explored in the algorithm satisfy the constraint $P(\mathrm{UE}) \leq \epsilon$. The retransmission probability in this case is the probability of decoding a NACK at the receiver, given by (3.66). The optimal transmission lengths $\{I_i\}^*$ obtained from the algorithm were evaluated in the two-phase, information-feedback setting for both the AWGN channel and BSC in Sec. 3.4.

An additional difference in the optimization of the two-phase increments for the BSC is that the communication-phase error probability predicted by RCSP is not monotonic in the blocklength. In order to avoid getting stuck at local optima, the two-phase version of Algorithm 1 for the BSC evaluated positive and negative length-2 steps of I_i in addition to unit steps. The confirmation-blocks N_{conf} searches were still limited to unit steps. We used $max\_restarts = 1000$ for the BSC and $max\_restarts = 100$ for the AWGN channel.

CHAPTER 4

Does Using Feedback Reduce Energy?

As shown by Polyanskiy et al. [25] and Chen et al. [49, 50], variable-length coding with feedback can significantly improve the maximum rate at short blocklengths compared to fixed-length codes without feedback. Equivalently, the latency required to achieve a certain rate close to capacity is lower for codes with feedback. When a transmitter's power consumption is proportional to the number of transmitted symbols, it is therefore tempting to conclude that variable-length feedback codes reduce power consumption compared to fixed-length codes at short blocklengths.

However, this conclusion ignores the impact of variable-length coding on the receiver's power consumption. In many applications, decoding energy constitutes a large portion of the overall energy budget. As discussed by Grover, Woyach, and Sahai in [78], the decoder power may be larger than the transmitter power for short-range communication. Whereas communication with fixed-length block codes requires decoding once per codeword, variable-length feedback codes require multiple decoding attempts. In Polyanskiy et al.'s VLF coding framework [25], decoding is attempted after every transmitted symbol, so an average blocklength of ℓ corresponds to ℓ decoding attempts (on average). Similarly, in [79], Polyanskiy et al. show that noiseless feedback dramatically improves the minimum transmitter energy required to send k bits through the Gaussian channel. The analysis in [79] permitted decoding after every symbol, but did not consider the associated receiver energy.

In contrast to [25] and [79], this chapter investigates the tradeoff between transmitter and receiver power consumption for VLF codes by analyzing an incremental redundancy scheme with uniform increments, which was introduced in [49, 50]. In this scheme, decoding occurs only after uniform increments of length $I \geq 1$. This limitation reduces the number of decoding attempts, similar to the m-transmission VLF codes of Ch. 3 and existing m-transmission schemes in the communication literature (e.g., [13, 15, 53]). Yet it is still not clear if the savings in transmitted symbols (and transmitter power) provided by feedback coding make up for the increase in decoding attempts (and receiver power) when compared to fixed-length coding.

To analyze this balance, Sec. 4.1 introduces an energy model that includes contributions from both the transmitter and receiver. Sec. 4.2 formulates an optimization problem for both fixed-length codes without feedback and VLF codes, and Sec. 4.3 concludes the chapter.

4.1 Energy Model

In order to fairly compare the impact of variable-length feedback coding on energy consumption, we use the following model. Transmission takes place over a binary-input AWGN channel with power constraint η (i.e., transmitted symbols are drawn from $\{+\sqrt{\eta}, -\sqrt{\eta}\}$). We assume w.l.o.g. that the noise variance is unity, so that the SNR is η. Due to the limited resources of the decoder, only hard-decision decoding is allowed, which means that the channel is effectively a binary symmetric channel (BSC). The Shannon capacity is $C_{\text{BSC}} = 1 - h_b(p)$, where $h_b(p)$ is the binary entropy and p is the crossover probability of the channel, related to η by $p = Q(\sqrt{\eta})$. The Q function $Q()$ is the tail of a standard normal random variable, i.e., $Q(x) = \frac{1}{\sqrt{2\pi}} \int\limits_x^\infty e^{-t^2/2} dt$.

The overall energy consumption $\mathcal{E}$ is given by

$$\mathcal{E} = \underbrace{\{\#\text{transmitted symbols}\} \times \alpha\eta}_{\text{Tx. energy}} + \underbrace{\{\#\text{ decoding attempts}\} \times \beta}_{\text{Rx. Energy}}. \tag{4.1}$$

In (4.1), inspired by similar formulas in [80] and [81], α is a constant related to the per-symbol energy consumption of the transmitter and β is a constant based on the per-codeword decoding energy. The parameter β may also be chosen to account for any other energy costs that are incurred for each decoding attempt, such as the wake-up/transition power consumed in the idle time between decoding attempts, for both the transmitter and the receiver. The energy required to send the decision-feedback message (ACK/NACK) may also be included in β. In this manner, many of the practical aspects of incremental redundancy that were ignored in the information-theoretic VLF framework of [25] can be accounted for by a proper selection of α and β. Both α and β are device-specific and depend on parameters such as the symbol time, decoding algorithm, etc. The units are chosen such that the energy $\mathcal{E}$ is in Joules.

Eq. (4.1) assumes that each decoding attempt uses the same blocklength, since otherwise the decoding energy may be less for smaller transmitted increments. This assumption is motivated by the incremental redundancy schemes in Ch. 3 that use punctured codes to decode an entire codeword when only a portion of the symbols have been transmitted.

For scenarios where the receiver power consumption is dominated by front-end amplifiers rather than the decoder, the receiver energy would scale with the number of transmitted symbols instead of the number of decoding attempts. In these cases, the benefits of feedback would be more obvious, since feedback can reduce the number of transmitted symbols. When the receiver power consumption has contributions that depend on both the number of transmitted symbols and the number of decoding attempts, both α and β can be chosen to include receiver contributions.

4.2 Optimization Problem

In order to compare the energy consumption of VLF and fixed-length codes without feedback, for both cases we assume that the transmitter uses a finite codebook with M messages and that messages are sent at regular intervals. The spacing of the transmission intervals is large enough that there is no queue (i.e., the transmission of each message finishes before the beginning of the next message's interval). For example, the transmitter may be a battery-operated sensor node that is reporting fixed-size environmental measurements every fifteen minutes.

4.2.1 Optimization for Fixed-length Codes without Feedback

For length-N block codes without feedback, the overall energy consumption $\mathcal{E}_{\text{no feedback}}$ is

$$\mathcal{E}_{\text{no feedback}} = N\alpha\eta + \beta, \tag{4.2}$$

since decoding occurs only once. Therefore, in order to to minimize the energy consumption, we must minimize the blocklength N (since α, β, and η are constants), subject to a constraint on the word-error probability ϵ:

$$\min \mathcal{E}_{\text{no feedback}}(N) = \mathcal{E}_{\text{no feedback}}(N^*), \tag{4.3}$$

$$N^* = \min\{N \in \mathbb{Z}^{++} : \exists\ (M, N, \epsilon)\text{-code}\}, \tag{4.4}$$

where Polyanskiy et al. [24] have defined an (M, N, ϵ) code as a codebook with M messages, N channel uses, and average probability of error less than ϵ. (This is the standard definition for finite-blocklength codes without feedback.)

In order to determine N^*, we can find the smallest N such that the maximum achievable finite-blocklength rate $\frac{\log M^*(N,\epsilon)}{N}$ is higher than the actual rate with M messages and blocklength N:

$$N^* = \min\left\{N \in \mathbb{Z}^{++} : \frac{\log M^*(N, \epsilon)}{N} \geq \frac{\log M}{N}\right\}. \tag{4.5}$$

In the numerical examples to follow, we have used the normal approximation of $M^*(N, \epsilon)$ from Polyanskiy et al. [24] to compute N^*. For the binary symmetric channel with crossover probability p, the maximim achievable finite-blocklength rate using the normal approximation is given as [24, Thm. 52]:

$$\frac{\log M^*(N, \epsilon)}{N} \approx C - \sqrt{\frac{p(1-p)}{N}} \log \frac{1-p}{p} Q^{-1}(\epsilon) + \frac{1}{2}\frac{\log N}{N}. \tag{4.6}$$

4.2.2 Optimization for Variable-length Feedback Codes

Similar to (4.2), the overall energy consumption $\mathcal{E}_{\text{VLF}}$ for variable-length feedback codes is

$$\mathcal{E}_{\text{VLF}} = \ \ell\alpha\eta + D\beta, \tag{4.7}$$

where ℓ is the average blocklength and D is the expected number of decoding attempts.

In the random-coding lower bound of Thm. 2, from Polyanskiy et al. [25], we have used an upper bound on ℓ to evaluate the average blocklength numerically:

$$\ell \leq \mathrm{E}[\tau], \tag{4.8}$$

$$\tau = \inf\{n \geq 0 : i(X^n; Y^n) \geq \gamma\}, \tag{4.9}$$

where τ is a stopping time defined according to (4.9) and γ is a threshold chosen to satisfy the error constraint ϵ.

In order to address the original concern for the increased receiver energy due to decoding after every individual symbol, we now introduce the following modified stopping time $\tau(I)$ for decoding interval $I \geq 1$ (from [49, 50]):

$$\tau(I) = \inf\{n \geq 0 : i(X^n; Y^n) \geq \gamma,\ n \in \mathcal{S}_I\}, \tag{4.10}$$

$$\mathcal{S}_I = \{n_0, n_0 + I, n_0 + 2I, \dots\} \tag{4.11}$$

where n_0 is the first blocklength for which decoding is attempted. That is, $\tau(I)$ represents the stopping time within the set of admissible blocklengths $\mathcal{S}_I$. Because the information density $i(X^n; Y^n)$ is not likely to be greater than the threshold γ until the instantaneous blocklength n is relatively close to capacity-achieving blocklength, it makes sense not to attempt decoding until $n \geq n_0$ for some constant n_0, hence the dependence on n_0 in the definition of $\mathcal{S}_I$.

In the numerical examples that follow, we have chosen n_0 according to

$$n_0 = \min\{n \geq 0 : \mathrm{P}[\tau > n] < 0.99\}, \tag{4.12}$$

although a variety of other heuristic choices of n_0 could be made. For example, Vakilinia et al. [65] select the first-attempt blocklength n_0 based on the empirical probability of decoding success for an LDPC code. For the channel parameters studied in this chapter, the choice of n_0 in (4.12) results in the first-attempt blocklength n_0 being smaller than the capacity-achieving blocklength $\frac{\log M}{C}$, which is essential for enabling VLF codes to achieve high rates.

With the definition of $\tau(I)$ in (4.10), the decoding energy $\mathcal{E}_{\text{VLF}}(I)$ can be upper bounded as follows:

$$\mathcal{E}_{\text{VLF}}(I) \leq \mathrm{E}[\tau(I)]\alpha\eta + \left(1 + \frac{\mathrm{E}[\tau(I)] - n_0}{I}\right)\beta, \tag{4.13}$$

since the average number of decoding attempts $D(I)$ is upper bounded as

$$D(I) \leq 1 + \frac{\mathrm{E}[\tau(I)] - n_0}{I}. \tag{4.14}$$

That is, decoding is always attempted at blocklength n_0, and then once per I transmitted symbols after that.

The optimization problem is now to choose the decoding interval I^* that minimizes the energy:

$$\min \mathcal{E}_{\text{VLF}}(I) = \mathcal{E}_{\text{VLF}}(I^*). \tag{4.15}$$

This optimization problem includes an implicit error constraint of ϵ, based on the threshold γ in (4.10), similar to the random-coding lower bound in Thm. 2.

Note the difference between the objective for the energy-minimization problem in (4.13) and the objective $\ell(I)$ for the related latency-minimization problem:

$$\ell(I) \leq \mathrm{E}[\tau(I)]. \tag{4.16}$$

In (4.16), choosing $I = 1$ minimizes the latency, but the optimal I^* in (4.13) is not obvious (and will likely have $I^* > 1$).

The latency-minimization objective in (4.16) is based on random coding and has been studied elsewhere for other types of feedback codes (e.g., [55, 56, 77]). Williamson et al.

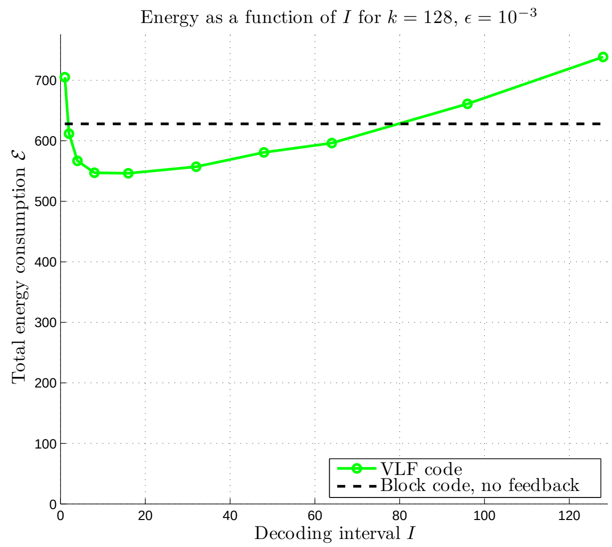


Figure 4.1: Total energy consumption as a function of the decoding interval I for variable-length feedback (VLF) codes over the BSC with crossover probability $p = 0.0565$, SNR $\eta = 4$ dB, and $\log_2 M = k = 128$ message bits. For both the VLF codes and the fixed-length block code without feedback, the word-error probability is required to be less than $\epsilon = 10^{-3}$.

optimized the latency for rate-compatible sphere-packing (RCSP) VLFT codes in [55], two-phase RCSP VLF codes in [56], and tail-biting convolutional VLF codes in [77], all in the context of m-transmission incremental redundancy schemes rather than the present scheme with uniform increments. This chapter uses the uniform increment approach because it is more general and permits optimization of only one parameter (i.e., the increment I).

4.2.3 Numerical Results

Fig. 4.1 illustrates the energy consumption $\mathcal{E}_{\text{VLF}}(I)$ as a function of the decoding interval I for the binary symmetric channel (BSC) with crossover probability 0.0565, SNR $\eta = 4$

dB, and $\log_2 M = k = 128$ message bits. As expected, overall energy consumption can be improved by decoding less often (i.e., with $I > 1$). Infrequent decoding with an increment I that is too large, however, increases the expected latency $\mathrm{E}[\tau(I)]$ substantially, increasing the transmitter power and offsetting the reduction in receiver power. In this example, the minimum energy consumption occurs for $I = 16$ (i.e., this is the minimum among the intervals that were evaluated).

Also shown in Fig. 4.1 is the minimum energy consumption $\mathcal{E}_{\text{no feedback}}(N^*)$ for block codes with $k = 128$ bits, where the blocklength N^* is computed as in (4.5). This energy consumption does not correspond to a particular value of I, but is plotted as a horizontal line to indicate how well the VLF codes fare in comparison to the no-feedback case. For both the VLF codes and fixed-length codes in Fig. 4.1, the word-error probability is required to be less than $\epsilon = 10^{-3}$.

Fig. 4.2 displays the energy consumption $\mathcal{E}_{\text{VLF}}(I)$ for different decoding intervals I as a function of the SNR in dB. For each SNR, the optimum decoding interval $I^*(\text{SNR})$ is plotted with a solid line and diamond marker. Although certain choices of decoding intervals I for VLF codes result in higher energy consumption than the fixed-length code at the same SNR, for each SNR there is at least one decoding interval $I^*(\text{SNR})$ that reduces the energy consumption compared to fixed-length coding.

In the numerical examples in this section, we used $\alpha = 1$ and $\beta = 2$. Increasing β would be expected to increase the optimum decoding interval I^* for VLF codes.

4.3 Concluding Remarks

Note that the results in Fig. 4.2 correspond to the binary symmetric channel, since we have assumed binary signaling (e.g., BPSK) over an AWGN channel and hard-decision decoding. SNRs are plotted on the x-axis in order to illustrate the connection between transmitter power η and overall energy consumption. The BSC crossover probability of $p = 0.0565$ in

Fig. 4.1 corresponds to an SNR of 4 dB (SNR $\eta = 2.5119$).

Another observation from Fig. 4.2 is that decreasing the SNR η can decrease the overall energy consumption required to transmit $k = 128$ bits with error probability less than $\epsilon = 10^{-3}$, even though decreasing the SNR reduces the achievable rate. This is expected from asymptotic information theory. Note that η is often treated as a constant in information theory, in order to optimize a particular quantity (e.g., the asymptotic capacity or the maximum achievable rate at finite blocklengths). The motivation usually given for η being a constant is that the transmitter has some fixed power constraint, based on factors such as its energy source (e.g., batteries), amplifiers, etc. In contrast, Fig. 4.2 illustrates that it is important to pick the transmitter power η wisely in order to minimize the energy consumption. In this manner, the battery life of a sensor node can be significantly extended over the device lifetime.

We conclude by noting that the maximum achievable rate for finite-blocklength codes without feedback is tightly upper bounded by the converse bounds given in [24]. For many channel parameters, the converses are well approximated by the normal approximation, which we have used in our numerical examples. It remains to be seen if practical codes exist both with rates close to the maximum and with energy-efficient decoding algorithms. In contrast, the random-coding lower bound for VLF codes from [25] is an achievability result that guarantees the existence of codes with (at least) the specified rates. We have shown in Ch. 3 that, indeed, (deterministic) tail-biting convolutional codes can deliver rates above the random-coding lower bound. In this section, we have not claimed the optimality of convolutional codes for reducing energy, nor have we investigated the actual decoding energy required for convolutional codes. Still, the conclusions from this section indicate that variable-length coding with feedback should be beneficial for reducing energy compared to the best fixed-length codes, but that choosing the appropriate decoding interval is crucial.

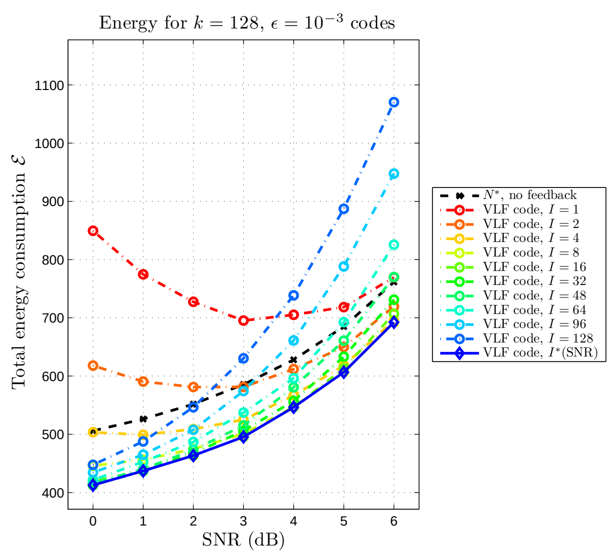


Figure 4.2: Total energy consumption as a function of the decoding interval I and the SNR for variable-length feedback (VLF) codes over the BSC with crossover probability $p = 0.0565$ and $\log_2 M = k = 128$ message bits. The solid line with diamond markers highlights the minimum energy across all values of I evaluated. For both the VLF codes and the fixed-length block code without feedback, the word-error probability is required to be less than $\epsilon = 10^{-3}$.

CHAPTER 5

Conclusion

This dissertation has demonstrated that feedback provides benefits in the short-blocklength regime, compared to fixed-length block codes without feedback. We offered several perspectives to support this conclusion, including new achievability bounds for VLF codes, based on random coding, as well as explicit constructions of coding schemes based on convolutional codes. The development of the tail-biting ROVA crucially enabled the use of throughput-efficient tail-biting convolutional codes in a reliability-based retransmission scheme. Finally, we compared the energy efficiency of codes with and without feedback.

Building on existing work in information theory that outlined the fundamental limits for VLF codes, we presented two improved lower bounds on the maximum rate at short blocklengths that use information feedback. Whereas the existing random-coding lower bound uses decision-feedback alone, our two-phase information-feedback bounds allow the transmitter to confirm or deny the receiver's tentative estimate in a confirmation phase. Inspired by a variety of two-phase schemes in the classical error-exponent literature that involve asymptotics, our work has focused on the performance of the scheme at blocklengths under 300 symbols. As expected intuitively, loosening the restriction from only allowing decision feedback to permitting information feedback improved the achievable rates.

We noted that although the confirmation phase in our schemes permits the transmitter to actively select coded symbols based on feedback, the transmitter is still relatively inactive, in the sense that it does not use feedback after every symbol. More frequent use of the feedback would be expected to lower latency further. In fact, existing research had introduced an active sequential hypothesis testing scheme that uses feedback after every symbol. While this scheme attains the optimal error-exponent, we showed that this scheme's lower bounds on rate are loose at short blocklengths, since they were designed for (asymptotic) error-exponent analysis. There are opportunities to improve these bounds and to design other active hypothesis testing schemes specifically for the finite-blocklength regime.

From the perspective of a practical communication system, decoding after every symbol is problematic and generally infeasible. With this in mind, we have shown how VLF codes can be transmitted in increments and decoded only at limited intervals, as in existing hybrid ARQ schemes. We have used insights from previous hybrid ARQ research to evaluate short-blocklength performance. Simulations of incremental redundancy schemes based on convolutional codes demonstrated that the restriction to limited decoding still permits rates above the random-coding lower bound. In particular, future systems with blocklengths under 100 symbols would benefit from the throughput efficiency of the TB ROVA, which does not suffer from rate penalties that occur with terminated convolutional codes and with CRCs. For blocklengths above 300 symbols, CRCs paired with alternative error-correction codes such as LDPC codes would likely be a better choice.

Our reliability-based retransmission schemes are the most practically relevant, since they use only a single bit of feedback per decoding attempt to convey the receiver's decision. This is true for the scheme based on the TB ROVA as well as for the code-based error-detection scheme using CRCs. While our two-phase incremental redundancy scheme used tail-biting convolutional codes and information feedback to deliver higher rates than the decision-feedback TB ROVA scheme, it is less likely to be adopted practically due to its increased feedback rate. Throughout this dissertation, we have assumed that the feedback link is noiseless, which is rarely the case. A single bit of feedback is often assumed to be noiseless when it can be communicated reliably by adding redundancy that imposes a negligible cost in the overall rate of the system. However, this assumption may fail to hold when feedback occurs often and when information feedback is used with large output alphabets.

Individual application scenarios must be evaluated to determine which feedback assumptions are valid. We introduced a general energy model that imposed a cost for repeated decoding attempts. The costs of feedback transmission could also be incorporated into this model. One possible application that may validate the noiseless feedback assumption is a communication link with asymmetry in power constraints or resources. For example, with a battery-operated sensor node transmitting to a base station, it may be reasonable to assume that the feedback rate is unlimited (and essentially noiseless).

There are many opportunities for further research of variable-length feedback codes with finite blocklengths, both in terms of information theory and code design. We have developed improved lower bounds on rate, but we have not addressed the converse bounds. Limiting the scope to special cases such as decision-feedback codes or specific channels may provide avenues to improve the converse bounds.

In our simulations, we used convolutional codes due to their excellent error performance at short blocklengths. Feedback codes with average latencies around 500 symbols, however, would benefit from other codes, such as LDPC codes. Focusing on specific codes may also enable specialized retransmission rules based on alternative reliability metrics, such as the likelihood ratios involved in iterative decoding. Further, alternative communication-and-confirmation schemes may provide benefits compared to those presented in this dissertation. Perhaps one of the most important goals for continuing work in this field is to design coding schemes with feedback assumptions that are practically relevant. In many cases, a system may be able to accommodate more than one bit of feedback per decoding attempt, but not unlimited-rate information-feedback.

References

[1] A. Raghavan and C. Baum, "A reliability output Viterbi algorithm with applications to hybrid ARQ," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 1214–1216, May 1998.

[2] M. Naghshvar, M. Wigger, and T. Javidi, "Optimal reliability over a class of binary-input channels with feedback," in *Proc. 2012 IEEE Inf. Theory Workshop (ITW)*, Sep. 2012, pp. 391–395.

[3] M. Naghshvar, T. Javidi, and M. A. Wigger, "Extrinsic Jensen-Shannon divergence: Applications to variable-length coding," submitted for publication. Available: http://arxiv.org/abs/1307.0067.

[4] P. Koopman and T. Chakravarty, "Cyclic redundancy code (CRC) polynomial selection for embedded networks," in *2004 IEEE Int. Conf. Dependable Systems and Networks (DSN)*, July 2004, pp. 145 – 154.

[5] S. Denic, "Robust incremental redundancy hybrid ARQ coding for channels with unknown interference," in *Proc. 2011 IEEE Int. Symp. Inf. Theory (ISIT)*, St. Petersburg, Russia, Aug. 2011, pp. 1658 –1662.

[6] I. Andriyanova and E. Soljanin, "IR-HARQ schemes with finite-length punctured LDPC codes over the BEC," in *2009 IEEE Inf. Theory Workshop (ITW)*, Taormina, Sicily, Oct. 2009, pp. 125 –129.

[7] R. Liu, P. Spasojevic, and E. Soljanin, "Punctured turbo code ensembles," in *2003 IEEE Inf. Theory Workshop (ITW)*, Paris, France, Mar. - Apr. 2003, pp. 249 – 252.

[8] Motorola, "Performance comparison of hybrid-ARQ schemes," 3GPP input paper TSGR1#17(00)1396, 2000.

[9] Lucent, "System performance comparison of Chase combining and adaptive IR for HSDPA," 3GPP input paper TSGR1#20(01)0587.

[10] P. Frenger, S. Parkvall, and E. Dahlman, "Performance comparison of HARQ with Chase combining and incremental redundancy for HSDPA," in *Proc. 54th IEEE Veh. Technol. Conf. (VTC)*, Oct. 2001, pp. 1829 –1833.

[11] J.-F. Cheng, "On the coding gain of incremental redundancy over Chase combining," in *Proc. 2003 IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2003, pp. 107 – 112.

[12] ——, "Coding performance of hybrid ARQ schemes," *IEEE Trans. Commun.*, vol. 54, no. 6, pp. 1017 –1029, June 2006.

[13] E. Visotsky, Y. Sun, V. Tripathi, M. Honig, and R. Peterson, "Reliability-based incremental redundancy with convolutional codes," *IEEE Trans. Commun.*, vol. 53, no. 6, pp. 987– 997, June 2005.

[14] E. Uhlemann, L. Rasmussen, A. Grant, and P.-A. Wiberg, "Optimal incremental-redundancy strategy for type-II hybrid ARQ," in *Proc. 2003 IEEE Int. Symp. Inf. Theory (ISIT)*, July 2003, p. 448.

[15] J. Fricke and P. Hoeher, "Reliability-based retransmission criteria for hybrid ARQ," *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2181–2184, Aug. 2009.

[16] H. Ma and J. Wolf, "On tail biting convolutional codes," *IEEE Trans. Commun.*, vol. 34, no. 2, pp. 104–111, Feb. 1986.

[17] S. Lin and D. J. Costello, *Error Control Coding, Second Edition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2004.

[18] J. Fricke and P. Hoeher, "Word error probability estimation by means of a modified Viterbi decoder," in *Proc. 66th IEEE Veh. Technol. Conf. (VTC)*, Oct. 2007, pp. 1113–1116.

[19] P. Ståhl, J. Anderson, and R. Johannesson, "A note on tailbiting codes and their feedback encoders," *IEEE Trans. Inf. Theory*, vol. 48, no. 2, pp. 529–534, Feb. 2002.

[20] H. Yamamoto and K. Itoh, "Viterbi decoding algorithm for convolutional codes with repeat request," *IEEE Trans. Inf. Theory*, vol. 26, no. 5, pp. 540–547, Sep. 1980.

[21] E. Hof, I. Sason, and S. Shamai (Shitz), "On optimal erasure and list decoding schemes of convolutional codes," in *Proc. Tenth Int. Symp. Commun. Theory and Applications (ISCTA)*, July 2009, pp. 6–10.

[22] G. Forney, "Exponential error bounds for erasure, list, and decision feedback schemes," *IEEE Trans. Inf. Theory*, vol. 14, no. 2, pp. 206–220, Mar. 1968.

[23] A. R. Williamson, T.-Y. Chen, and R. D. Wesel, "Reliability-based error detection for feedback communication with low latency," in *Proc. 2013 IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, July 2013.

[24] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307 –2359, May 2010.

[25] ——, "Feedback in the non-asymptotic regime," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 4903–4925, Aug. 2011.

[26] J. B. Anderson and S. M. Hladik, "Tailbiting MAP decoders," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 297–302, Feb. 1998.

[27] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[28] J. B. Anderson and K. E. Tepe, "Properties of the tailbiting BCJR decoder," in *Codes, Systems, and Graphical Models*, B. Marcus and J. Rosenthal, Eds. New York: Springer-Verlag, 2001, pp. 211–238.

[29] J. B. Anderson and S. M. Hladik, "An optimal circular Viterbi decoder for the bounded distance criterion," *IEEE Trans. Commun.*, vol. 50, no. 11, pp. 1736–1742, Nov. 2002.

[30] R. Johannesson and K. Zigangirov, *Fundamentals of convolutional coding.* Piscataway, NJ, USA: IEEE Press, 1999.

[31] M. Handlery, R. Johannesson, and V. Zyablov, "Boosting the error performance of suboptimal tailbiting decoders," *IEEE Trans. Commun.*, vol. 51, no. 9, pp. 1485–1491, Sep. 2003.

[32] N. Y. Yu, "Performances of punctured tail-biting convolutional codes using initial state estimation," in *Proc. 68th IEEE Veh. Technol. Conf. (VTC)*, Sep. 2008, pp. 1–5.

[33] R. Cox and C.-E. W. Sundberg, "An efficient adaptive circular Viterbi algorithm for decoding generalized tailbiting convolutional codes," *IEEE Trans. Veh. Technol.*, vol. 43, no. 1, pp. 57–68, Feb. 1994.

[34] T.-Y. Wu, P.-N. Chen, H.-T. Pai, Y. S. Han, and S.-L. Shieh, "Reliability-based decoding for convolutional tail-biting codes," in *Proc. 71st IEEE Veh. Technol. Conf. (VTC)*, May 2010.

[35] R. Y. Shao, S. Lin, and M. P. C. Fossorier, "Two decoding algorithms for tailbiting codes," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1658–1665, 2003.

[36] H.-T. Pai, Y. S. Han, and Y.-J. Chu, "New HARQ scheme based on decoding of tail-biting convolutional codes in IEEE 802.16e," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 912–918, Mar. 2011.

[37] P. Shankar, P. N. A. Kumar, K. Sasidharan, and B. Rajan, "ML decoding of block codes on their tailbiting trellises," in *Proc. 2001 IEEE Int. Symp. Inf. Theory (ISIT)*, Washington, DC, June 2001, p. 291.

[38] P. Shankar, P. N. A. Kumar, K. Sasidharan, B. S. Rajan, and A. S. Madhu, "Efficient convergent maximum likelihood decoding on tail-biting trellises," 2006. Available: http://arxiv.org/abs/cs/0601023.

[39] J. Ortín, P. García, F. Gutiérrez, and A. Valdovinos, "A* based algorithm for reduced complexity ML decoding of tailbiting codes," *IEEE Commun. Lett.*, vol. 14, no. 9, pp. 854–856, Sep. 2010.

[40] J. Anderson and K. Balachandran, "Decision depths of convolutional codes," *IEEE Trans. Inf. Theory*, vol. 35, no. 2, pp. 455–459, Mar. 1989.

[41] C. E. Shannon, "The zero error capacity of a noisy channel," *IRE Trans. Inf. Theory*, vol. 2, no. 3, pp. 8–19, Sep. 1956.

[42] C. Suh and D. Tse, "Feedback capacity of the Gaussian interference channel to within 2 bits," *IEEE Trans. Inf. Theory*, vol. 57, no. 5, pp. 2667–2685, May 2011.

[43] D. Mandelbaum, "An adaptive-feedback coding scheme using incremental redundancy (corresp.)," *IEEE Trans. Inf. Theory*, vol. 20, no. 3, pp. 388 – 389, May 1974.

[44] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.

[45] R. L. Dobrushin, "Asymptotic bounds on error probability for transmission over DMC with symmetric transition probabilities," *Theory Probab. Appl.*, vol. 7, pp. 283–311, 1962.

[46] M. V. Burnashev, "Data transmission over a discrete channel with feedback. Random transmission time," *Probl. Inf. Transm.*, vol. 12, no. 4, pp. 10–30, 1976.

[47] ——, "Sequential discrimination of hypotheses with control of observation," *Math. USSR, Izvestia*, vol. 15, no. 3, pp. 419–440, 1980.

[48] J. Schalkwijk and T. Kailath, "A coding scheme for additive noise channel with feedback–I: No bandwidth constraint," *IEEE Trans. Inf. Theory*, vol. 12, no. 2, pp. 172–182, Apr. 1966.

[49] T.-Y. Chen, A. R. Williamson, and R. D. Wesel, "Variable-length coding with feedback: Finite-length codewords and periodic decoding," in *Proc. 2013 IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, July 2013.

[50] T.-Y. Chen, A. R. Williamson, N. Seshadri, and R. D. Wesel, "Feedback communication systems with limitations on incremental redundancy," submitted for publication. Available: http://arxiv.org/abs/1309.0707.

[51] T.-Y. Chen, N. Seshadri, and R. D. Wesel, "A sphere-packing analysis of incremental redundancy with feedback," in *Proc. 2011 IEEE Int. Conf. Commun. (ICC)*, Kyoto, Japan, June 2011.

[52] ——, "Incremental redundancy: A comparison of a sphere-packing analysis and convolutional codes," in *2011 Inf. Theory and Applications Workshop (ITA)*, San Diego, CA, USA, Feb. 2011.

[53] C. Lott, O. Milenkovic, and E. Soljanin, “Hybrid ARQ: Theory, state of the art and future directions,” in *2007 IEEE Inf. Theory Workshop for Wireless Networks*, Bergen, Norway, July 2007.

[54] A. R. Williamson, M. J. Marshall, and R. D. Wesel, “Reliability-output decoding of tail-biting convolutional codes,” *IEEE Trans. Commun.*, to be published. Available: http://arxiv.org/abs/1312.1024.

[55] A. R. Williamson, T.-Y. Chen, and R. D. Wesel, “A rate-compatible sphere-packing analysis of feedback coding with limited retransmissions,” in *Proc. 2012 IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, MA, USA, July 2012.

[56] ——, “Firing the genie: Two-phase short-blocklength convolutional coding with feedback,” in *2013 Inf. Theory and Applications Workshop (ITA)*, San Diego, CA, USA, Feb. 2013.

[57] A. Sahai, “Why do block length and delay behave differently if feedback is present?” *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 1860–1886, May 2008.

[58] S. Chang, “Theory of information feedback systems,” *IRE Trans. Inf. Theory*, vol. 2, no. 3, pp. 29–40, Sep. 1956.

[59] J. L. Massey, “Causality, feedback and directed information,” in *Proc. 1990 IEEE Int. Symp. Inf. Theory and its Applicat. (ISITA)*, Honolulu, Hawaii, USA, Nov. 1990.

[60] M. Naghshvar and T. Javidi, “Active sequential hypothesis testing,” *Ann. of Stat.*, vol. 41, no. 6, pp. 2703–2738, Dec. 2013.

[61] ——, “Extrinsic Jensen-Shannon divergence with application in active hypothesis testing,” in *Proc. 2012 IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, MA, USA, July 2012, pp. 2191–2195.

[62] ——, “Optimal reliability over a DMC with feedback via deterministic sequential coding,” in *Proc. 2012 IEEE Int. Symp. Inf. Theory and its Applicat. (ISITA)*, Honolulu, HI, USA, Oct. 2012, pp. 51–55.

[63] B. Mertz, J. Heide, J. Sorensen, R. Krigslund, and S. Pedersen, “Communication beyond (N)ACKs: Wireless transmission with informative feedback,” in *2008 Annu. IEEE Student Paper Conf.*, Feb. 2008, pp. 1–4.

[64] J. Freudenberger and B. Stender, “An algorithm for detecting unreliable code sequence segments and its applications,” *IEEE Trans. Commun.*, vol. 52, no. 11, pp. 1833–1839, Nov. 2004.

[65] K. Vakilinia, T.-Y. Chen, S. V. S. Ranganathan, A. R. Williamson, D. Divsalar, and R. D. Wesel, "Short-blocklength non-binary LDPC codes with feedback-dependent incremental transmissions," in *Proc. 2014 IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, July 2014.

[66] J. Ooi and G. W. Wornell, "Fast iterative coding for feedback channels," in *Proc. 1997 IEEE Int. Symp. Inf. Theory (ISIT)*, June 1997, p. 133.

[67] ——, "Fast iterative coding techniques for feedback channels," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 2960–2976, Nov. 1998.

[68] J. M. Ooi, *Coding for channels with feedback.* Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[69] R. G. Gallager, *Stochastic Processes: Theory for Applications.* Cambridge, UK: Cambridge Univ. Press, 2013.

[70] A. Wald, *Sequential Analysis.* New York, NY, USA: John Wiley & Sons Inc, 1947.

[71] T.-Y. Chen, A. R. Williamson, and R. D. Wesel, "Asymptotic expansion and error exponent of two-phase variable-length coding with feedback for discrete memoryless channels," in *Proc. 2014 IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, July 2014.

[72] M. Naghshvar and T. Javidi, "Sequentiality and adaptivity gains in active hypothesis testing," *arXiv*, 2012. Available: http://arxiv.org/abs/1211.2291.

[73] A. O. Nasif and G. N. Karystinos, "Binary transmissions over additive Gaussian noise: A closed-form expression for the channel capacity," in *Proc. 2005 Conf. Inf. Sci. and Syst. (CISS)*, Baltimore, MD, USA, Mar. 2005.

[74] K. Witzke and C. Leung, "A comparison of some error detecting CRC code standards," *IEEE Trans. Commun.*, vol. 33, no. 9, pp. 996–998, Sep. 1985.

[75] T.-Y. Chen, D. Divsalar, and R. D. Wesel, "Chernoff bounds for analysis of rate-compatible sphere-packing with numerous transmissions," in *Proc. 2012 IEEE Inf. Theory Workshop (ITW)*, Lausanne, Switzerland, Sep. 2012.

[76] C. E. Shannon, "Probability of error for optimal codes in a Gaussian channel," *Bell Syst. Tech. J.*, vol. 38, no. 2, pp. 611–656, 1959.

[77] A. R. Williamson, T.-Y. Chen, and R. D. Wesel, "Variable-length convolutional coding for short blocklengths with decision feedback," submitted for publication.

[78] P. Grover, K. Woyach, and A. Sahai, "Towards a communication-theoretic understanding of system-level power consumption," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1744–1755, Sept. 2011.

[79] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Minimum energy to send k bits through the Gaussian channel with and without feedback," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 4880 –4902, Aug. 2011.

[80] Y. Sankarasubramaniam, I. Akyildiz, and S. McLaughlin, "Energy efficiency based packet size optimization in wireless sensor networks," in *Proc. 2003 IEEE Int. Workshop on Sensor Netw. Protocols and Applicat.*, 2003, pp. 1–8.

[81] I. Stanojev, O. Simeone, Y. Bar-Ness, and D. H. Kim, "Energy efficiency of non-collaborative and collaborative hybrid-ARQ protocols," *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, pp. 326–335, Jan. 2009.