Title

Feedback Communication over the Binary Symmetric Channel: Analytical Bounds and Encoding Techniques

Author

Antonini, Amaael

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Feedback Communication over the Binary Symmetric Channel: Analytical Bounds and Encoding Techniques

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

by

Amaael Antonini

2024

ABSTRACT OF THE DISSERTATION

Feedback Communication over the Binary Symmetric Channel:
Analytical Bounds and Encoding Techniques

by

Amaael Antonini
Doctor of Philosophy in Electrical & Computer Engineering
University of California, Los Angeles, 2024
Professor Richard D. Wesel, Chair

This dissertation investigates communication over the binary symmetric channel with noiseless feedback of the received symbols from the decoder to the encoder. The binary symmetric channel receives as input a binary symbol, and produces as output also a binary symbol, that may be different to the input symbol. The channel is symmetric in the sense that the probability that the output symbol is not equal to the input symbol is the same for both input symbols. The general communication problem consists of reliably relaying a message from a source (where it is generated) to a destination (where it is needed) while using the least possible system resources. Reliability in this case is defined as a small error probability–the probability that the message at the destination differs from that at the source. System resources include forward channel symbol transmissions, encoder and decoder complexity, and possibly other resources that may be associated with a cost, like the number of times the feedback channel is accessed for a feedback transmission. The system operates as follows: the source delivers its message to an encoder, that computes channel input symbols; the channel produces output symbols that are noisy versions of the input symbols; a decoder

uses channel output symbols to compute an estimate of the message at the encoder; and a noiseless feedback channel delivers the output symbols to the encoder, which affords the encoder all the information available to the decoder. The objectives of this dissertations are (1) to design encoding methods that can be implemented in simulations; (2) explore simplifications that make the implementations more efficient; (3) analyze the expected rate that can be achieved with all the proposed encoding methods, including the simplifications; (4) analyze converse bounds, the lowest rates that cannot be achieved with the system while satisfying an error probability constraint; and (5) explore communication with additional constraints, like source causality constraints and feedback sparsity constraints.

Chapter 1 lays out the background and motivation of the research problems studied in this dissertation, and summarizes previous results and about related problems. Then, Ch. 1 briefly summarizes the contents of each Chapter.

Chapter 2 provides efficient algorithms and a simulation framework that implements previously proposed encoders as well as new ones. Simulation results validate previous achievability bounds and motivate tighter achievability bounds. Chapter 2 also proposes a simpler encoding rule, that greatly lowers the runtime and memory complexity, and exhibits a rate performance indistinguishable from the encoders with the highest existing achievability bounds. The performance of the new encoder is further analyzed in Ch. 3.

Chapter 3 demonstrates a new analysis of the expected blocklength needed to transmit a fixed length information sequence with bounded error probability. The new analysis relaxes the sufficient encoding constraints that guarantee an expected rate performance above the highest achievability bounds previously developed for the model. To tighten an upper bound on expected time, analyzed for the first phase of a two-phase process, this chapter proposes an analysis of a "surrogate process," for which a tighter bound can be shown. The "surrogate process" is carefully constructed from the original process, so that its decoding time upper bounds that of the original process. This property guarantees that the tighter bound on the "surrogate process" applies to the original, and results in a significantly higher achievability

bound. The bounds are tightened further by jointly optimizing both phases of the two-phase analysis used to obtain the original bounds. A proof that the simple encoder of Ch. 2 satisfies the relaxed constraints is provided. Chapter 3 then proposes a converse bound, an upper bound on the highest expected rate that is achievable by an encoder that enforces the stopping rule used by the proposed encoders.

Chapter 4 extends the study of feedback codes over the binary symmetric channel to the "causal encoding" setting, where the source information sequence becomes progressively available during transmission instead of the traditional setting where the entire source sequence is available before transmission begins. In "causal encoding" the encoder seeks to minimize the average decoding delay under the same frame error rate constraint considered in Ch. 2 and Ch. 3. The sub-block combining algorithm is proposed as a "causal encoder," that starts the transmission with a segment of the information sequence, and adds new segments as they become available. The chapter identifies lower bounds on expected decoding times imposed by the channel and the source, as well as the region where a causal encoder may outperform existing non-causal encoders. Simulation results are provided to show that the sub-block combining algorithm outperforms, in average decoding time, non-causal encoders, in their original form, and with natural modifications that make them better "causal encoders" under certain conditions. The performance of the sub-block combining algorithm is further improved using a method that analyzes optimized block sizes for the specific operating point, set by the source and channel symbol rates.

Chapter 5 studies the "sparse feedback" setting where feedback symbols are only available to the transmitter at sparse time instances, instead of being available before the transmission of each symbol. A new encoding rule is introduced, that also satisfies the relaxed constraints proposed in Ch. 3. Then the "look-ahead algorithm" is proposed, to satisfy the encoding constraints for a few transmissions in advance, without additional feedback. Simulation results show that the "look-ahead algorithm" admits average feedback delay that increases with message length, from slightly above one transmission at a message length of about ten

bites, to about five to six when the message size reaches 80 bits. The sub-block combining algorithm of Ch. 4 is modified to operate on several blocks simultaneously and used as a sparse feedback encoder. This algorithm exhibits much lower complexity, which makes it suitable for message sizes of up to a few hundred bits, that are not possible with the look-ahead algorithm.

Chapter 6 provides the conclusions of this dissertation, and highlights future research directions.

The dissertation of Amaael Antonini is approved.

Lara Dolecek

Danijela Cabric

Dariush Divsalar

Alexander Sherstov

Richard D. Wesel, Committee Chair

University of California, Los Angeles

2024

To my parents and brothers

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I want to express my deepest gratitude to my Ph. D. advisor Professor Richard D. Wesel for his unwavering support and guidance during both my time in graduate school and as an undergraduate student at UCLA. To Professor Wesel, you supported me when I needed it most. Starting from Bruin Day after my admittance, your hands-on support when I faced logistical difficulties made me feel welcomed and at home at UCLA. Thank you for supporting my decision to start graduate school, and for your support when I needed guidance in my research, writing and publishing of conference and journal papers, as well as on many other occasions too numerous to mention.

I want to thank Professor Dariush Divsalar for his invaluable collaboration and guidance throughout my research and projects at UCLA's communication systems laboratory (CSL). In particular, Professor Divsalar assisted me with improving the decoder of the SCPPM code in the CCSDS standard used by JPL for deep space communication, and has also generously shared his extensive knowledge and experience in communication theory and its applications, which greatly benefited many projects, including my own, during my time in grad-school at UCLA. Additionally, Professor Divsalar was a member in my Ph.D. committee.

I want to thank the members of my doctoral committee Prof. Lara Dolecek, Prof. Danijela Cabric, Prof. Dariush Divsalar, Prof. Alexander Sherstov and Prof. Richard D. Wesel for lending their time to administer both my Oral Qualification Exam and Final Oral Exam, as well as for reviewing my research and dissertation. I know how challenging it is to allocate additional time for these activities, and I very much appreciate the dedication and commitment of each member of my doctoral committee in supporting me through the successful completion of the Ph.D. program at UCLA. Prof. Danijela Cabric was also my instructor for a class in Wireless Communication (ECE233) in Spring 2022, where she helped enhance my understanding of communication systems, extending well beyond channel coding. The knowledge I acquired in Professor Cabric's class proved invaluable during an internship where

I analyzed a wireless network for controlling large numbers of robots at Symbotic LLC, where I now work.

I want to thank Deeona Columbia for her outstanding work as the manager of graduate students affairs. I also want to thank Mary Ann Gerber and Alina Haas whom provided me assistance as an undergraduate student at UCLA. I feel humbled by the efficiency and dedication with which the engineering school is managed. It does not go unnoticed.

I want to thank Professor Dana Watson for being an outstanding writing instructor. For the first time, I felt my writing skills and style really improved through her Technical Writing course.

I want to thank Dr. Hengjie Yang, Dr. Linfang Wang, and Ph.D. students Semira Galijasevic, Wenhui (Beryl) Sui and Ava Asmani, whom I conducted research and worked closely with at UCLA's communication systems laboratory. Dr. Yang was the most senior student in Professor Wesel's research group when I joined, and developed many of the results described in this dissertation.

I want to thank Rita Gimelshein for her outstanding work coding the algorithms, running simulations and producing many of the graphs used in Ch. 4 and Ch. 5 of this dissertation.

I want to thank Zihan (Bruce) Qu and Laura Huang with whom I had the pleasure to collaborate with in research projects.

I want to thank all the professors at UCLA for their commitment and dedication that make UCLA the great school we know. I also want take this opportunity to thank all the staff at UCLA for their contribution and great work keeping the school running.

I want to thank Semira Galijasevic, Sharon Zhen and Dr. Wang for their help in formatting, editing and improving this dissertation and the presentation slides.

I want to thank Professor Yury Polyanskiy for kindly providing the data to plot some of the bounds he developed for related work on a stop feedback system.

I want to use this opportunity to thank Ozarks Technical Community College (OTC)

Chancellor Dr Hal L. Higdon, my advisor at OTC Professor Todd van Gorden and Professor Gary King for their support during my time there.

I also want to thank my professor and teachers in Venezuela, Professors Jose Villa, Professor Ernesto Gimenez, Professor Jesus A. Ocando, late Professor Ralsi C. Ocando, Professor Walter Wenzel, Professor Jose I. de la Cruz, Liliana de la Cruz, my aunts Deborah Seba and Marielvira Seba, my uncle Pablo Seba, and my late uncle Luigi Antonini. They helped me significantly to get to where I am and I will always remember the great times we shared.

Finally I want to thank my family for their support during my time in graduate school. My mother Susana Seba, my father Ronald Antonini, my siblings Benjamin, Amado de J. Ronald J. Obed and Lawrence. They were there for me all the time and helped me make it through grad school.

VITA

2020,2021	AI Engineering Intern, Aira Technologies.
2020	Teaching Assistant, University of California, Los Angeles
2023	Graduate Student Researcher, University of California, Los Angeles
2023	Engineering Intern, Symbotic
2024	Teaching Assistant, University of California, Los Angeles

PUBLICATIONS

A. Antonini, R. Gimeshein, and R. D. Wesel, "Achievable Rates for Low-Complexity Posterior Matching over the Binary Symmetric Channel," in *IEEE Transactions on Information Theory (TIT)*, 2023.

A. Antonini, H. Yang and R. D. Wesel, "Low Complexity Algorithms for Transmission of Short Blocks over the BSC with Full Feedback," in *IEEE International Symposium on Information Theory (ISIT)*, Los Angeles, CA, USA, June 21-26, 2020, pp. 2173-2178, doi: 10.1109/ISIT44484.2020.9174232.

A. Antonini, R. Gimelshein and R. D. Wesel, "Causal (Progressive) Encoding over Binary Symmetric Channels with Noiseless Feedback," in *IEEE International Symposium on Information Theory (ISIT)*, Melbourne, Australia, July 12-20, 2021, pp. 142-147, doi: 10.1109/ISIT45174.2021.9518287.

A. Antonini, W. Sui, B. Towell, D. Divsalar, J. Hamkins and R. D. Wesel, "Suppressing Error Floors in SCPPM via an Efficient CRC-aided List Viterbi Decoding Algorithm," in *12th International Symposium on Topics in Coding (ISTC)*, Brest, France, September 04-08, 2023, pp. 1-5, doi: 10.1109/ISTC57237.2023.10273548.

A. Antonini, and R. D. Wesel, "Feedback Communication Over the Binary Symmetric Channel with Sparse Feedback Times," in *In 2024 IEEE Global Communications Conference (GLOBECOM)*, Cape Town, South Africa, December 08-12, 2024.

R. Wesel, **A. Antonini**, L. Wang, W. Sui, B. Towell and H. Grissett, "ELF Codes: Concatenated Codes with an Expurgating Linear Function as the Outer Code," in *12th International Symposium on Topics in Coding (ISTC)*, Brest, France, September 04-08, 2023, pp. 1-5, doi: 10.1109/ISTC57237.2023.10273535.

Zihan Qu, **A. Antonini**, W. Sui, E. Min, A. Yang, and R. D. Wesel, "Complementary Exclusion of Full Polynomials to Enable Dual List Decoding of Convolutional Codes," in *IEEE Transactions on Information Theory (ISIT)*, Athens, Greece, June 07-12, 2024.

H. Yang, M. Pan, **A. Antonini**, and R. D. Wesel, "Sequential Transmission Over Binary Asymmetric Channels With Feedback," in *IEEE Transactions on Information Theory (TIT)*, vol. 68, no. 11, pp. 7023-7042, Nov. 2022, doi: 10.1109/TIT.2022.3179656.

CHAPTER 1

Introduction

This dissertation studies feedback communication over the binary symmetric channel (BSC) with noiseless feedback. The BSC is a simple model for communication that reveals the essential role that feedback can play at short blocklength. Feedback communication over the BSC has been widely studied over the years. Shannon [Sha48] showed that feedback does not increased the capacity of discrete memoryless channels, including the BSC. However, Shannon's capacity is an asymptotic limit as the blocklength goes to infinity. For finite blocklengths, and especially for short blocklengths, feedback has been shown to increase the decay rate of the error probability, e.g. by Burnashev [Bur76], when variable length codes are used. Many transmission methods have been developed for feedback communication over the BSC. An early well-known scheme was introduced by Horstein [Hor63], and subsequently generalized as "posterior matching" by [SF11]. Naghshvar *et al.* [NJW15] introduced a very efficient posterior matching scheme, which is studied in detail in this dissertation. Achievability and converse bounds have also been developed for the BSC with feedback limited to "stop feedback," like Polyanskiy's VLF [PPV11] , and with full noiseless feedback by Yang *et al.* [YPA21].

The system model consists of an information source, an encoder, a forward noisy channel, a noiseless feedback channel and a decoder, see Fig. 1.1. The source generates a binary sequence that is received by the encoder. The encoder generates binary transmitter symbols using the source information sequence and the feedback from the noiseless feedback channel. The forward channel, the BSC, produces binary channel symbols according to the

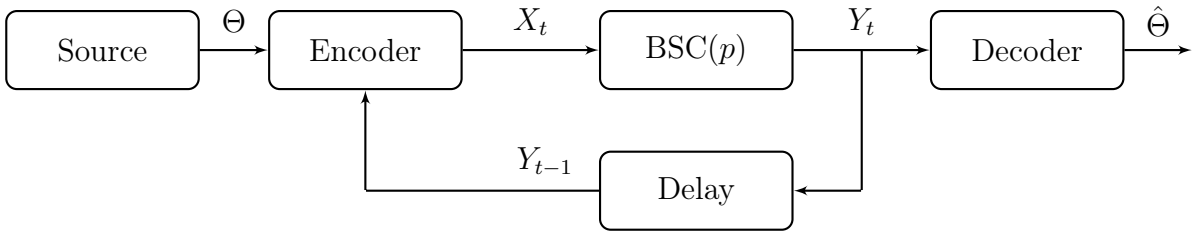

Figure 1.1: System diagram for communication over the BSC with a noiseless feedback channel. The information source generates a binary information sequence Θ; the encoder receives the information sequence and encodes, at each time t, a binary channel symbols X_t that is transmitted over the binary symmetric channel. the binary symmetric channel (BSC), with crossover probability p, produces binary output symbols Y_t, $t = 1, 2, \ldots$, that are noisy versions of the input symbols X_t, $t = 1, 2, \ldots$, according to the channel transition function: $\Pr(Y_t \neq X_t) = p$. The channel output symbols Y_t are sent back to the encoder through the noiseless feedback channel, and may be used by the encoder to encode the next symbol X_{t+1}. The decoder receives the channel symbols and, at the end of the transmissions, produces an estimate $\hat{\Theta}$ of the source information sequence Θ.

channel transition function. The channel transition function is a random function that produces a symbol different from the transmitted symbol according to the channel's crossover probability. The decoder receives the channel symbols and, at the end of the transmission, produces an estimate of the transmitted message, the source information sequence. The feedback channel delivers the channel's output symbols to the encoder, which allows the encoder access to the same information available to the decoder.

The BSC is a special case of the broader class of discrete memoryless channels. These channels have discrete (countable) input and output alphabets, and have a channel transition function that governs the probability that an output symbol is received given the symbol that is transmitted. In memoryless channels, the channel has no "state." The transition function is fixed. The output symbol depends only on the input symbol and the transition function, not on the time, and not on previous channel inputs or outputs. There are different symmetries for discrete memoryless channels. For the BSC the symmetry consists on equal probability that either input symbol is transformed into the other symbol by the channel.

The noiseless feedback channel affords the encoder all the information available to the

decoder, in addition to the information sequence. The highest achievable performance of this model upper bounds the performance of the same BSC with no feedback or any other limited feedback, like stop feedback [PPV10] for given frame error rate constraints. This becomes clear by noting that the noiseless feedback model can be downgraded to any other feedback model by restricting, or ignoring, the feedback used to encode each symbol. For instance, the encoder could make a "copy" of the decoder, that only signals the end of the process, thus transforming full and noiseless feedback into stop feedback, also noiseless.

Horstein [Hor63] developed a well known variable length schemes for the BSC with noiseless feedback. In Horstein's scheme, every possible message is represented as a sub-interval in a unit interval $(0, 1)$. The transmitter divides the unit interval into two equal halves and transmits 0 if the interval containing the transmitted message is in the first half, otherwise it transmits a 1. Let the channel's crossover probability be p and let $q = 1 - p$. Then, the received symbol is used to adjust the sizes of the sub-intervals on each half by $2q$ and $2p$ depending on what symbol was received. Both encoder and decoder do this, since the encoder gets access to the received symbol via the noiseless feedback channel. Horstein's scheme terminates when a sub-interval grows sufficiently large, according to a pre-selected decoding threshold. Horstein's sequential transmission scheme was presumed to achieve the capacity of the BSC. Shayevitz and Feder [SF11] analyzed a class of encoding schemes, which they call "posterior matching" (PM) schemes, that achieve the capacity of discrete memoryless channels . Shayevitz and Feder defined a "posterior matching" scheme as one that satisfies the "posterior matching principle" described as follows:

1. The channel input symbol X_{t+1}, at time $t + 1$ is a fixed function of a random variable U, that is independent of the received symbol history $Y^t \triangleq \{Y_1, Y_2, \ldots, Y_t\}$;

2. The transmitted message, Θ, can be uniquely recovered from (U, Y^t) a.s.

Shayevitz and Feder [SF11] then proved that Horstein's scheme achieves the capacity of the BSC, by showing that it satisfies the "posterior matching principle" and thus is a "posterior

matching" scheme. Gorantla and Coleman [GC10] used Lyapunov functions for an alternative proof that PM schemes achieve the channel capacity. A notable fixed length "Posterior Macching" scheme for discrete memoryless channels was introduced by Li and El-Gamal [LG15]. Their scheme used a random cyclic shift that was later used by Shayevitz and Feder for a simpler proof that Horstein's scheme achieves the capacity of the BSC [SF16]. Naghshvar *et al.* [NJW15] proposed a variable length, single phase "posterior matching" scheme for discrete DMC channels with feedback that exhibits Burnashev's optimal error exponent, and used a sub-martingale analysis to prove that it achieves the channel capacity. Polyanskyi [PPV10] developed a rate lower bound as a function of blocklength for variable length, stop feedback systems. In a stop feedback system, after every transmitted symbol the receiver sends the transmitter a true/false symbol that signals the end of the transmission, or equivalently, there is a single feedback symbol that is sent by the receiver when it determines the end of the transmission. Bae and Anastasopoulos [BA10] proposed a "posterior matching" scheme that achieves the capacity of finite state channels with feedback. Since then, other "posterior matching" algorithms have been developed, see [KPV17, KMM13, SPK18, Tru14, Ana12]. Other variable length schemes that attain Burnashev's optimal error exponent have also been developed, and some can be found in [Sch71, SP73, TT02, TT06, NWJ12].

Chapter 2 provides efficient posterior-matching algorithms and corresponding simulation results for transmission over the BSC with noiseless feedback under a frame error rate constraint. The simulation results validate recent achievability bounds, such as the SED bound by Yang *et al.* [YPA21], developed for Naghshvar's encoder [NJW15]. However, the simulation results also reveal a wide gap between actual performance and previous achievability bounds, which motivates the development of tighter bounds. The simulation results include a lower complexity algorithm that is facilitated by a relaxation of constraints that is justified in Chapter 3.

Chapter 3 explores a new analysis showing that the bounds developed by Yang *et al.* for the small-enough distance (SED) posterior-matching encoder developed by Naghshvar

et al. [NJW15] can be achieved with a simpler encoder facilitated by relaxing the SED constraint. The relaxed constraint does not admit the submartingale analysis of Yang *et al* [YPA21], but the new analysis replacing that submartingale analysis allows for even tighter achievability bounds. The new analysis also incorporates and generalizes the use of a surrogate process [YPA21], synthesized from the original process, to further tighten the achievability bounds. The surrogate process avoids the paradox where a performance improvement (an acceleration of convergence at the end of the transmission) loosens the achievability bound.

Chapter 4 studies causal encoding, which enforces on the encoder an information source availability constraint where information arrives at the encoder throughout the encoding process. The encoder does not have full knowledge of the message at the start of the transmission, but rather can only use information causally as the transmission progresses. The goal of causal encoding is to minimize the expected decoding time starting when the source delivers the first message bit to the encoder. Instead of waiting for the arrival of the entire information sequence before starting the transmission, the causal encoder must make efficient use of every opportunity to transmit a symbol, starting when the first information symbol arrives. It is also desired to maintain a manageable encoder complexity to make the implementation feasible. For a causal encoder, this dissertation proposes a sub-block combining algorithm that begins transmitting when a small message segment is available and combines new segments as they arrive. This dissertation identifies the operating regions where a causal encoder can outperform existing non-causal encoders. Analytical results include quantifying the decoding latencies achievable by causal and non-causal decoders and the gap between these two approaches. The segment sizes, or sub-block sizes, are optimized to reduce the decoding latency. Simulation results demonstrate that the sub-block combining algorithm significantly lowers the average decoding time over non-causal encoders, and performs close to the bounds inherent to the system over a wide portion of the region.

Chapter 5 proposes a sparsity constraint on feedback communication over the BSC.

The constraint limits the number of times the feedback channel is accessed by the receiver, not the number of symbols that are transmitted on each access. All the received symbols eventually become available to the encoder, but they do not all arrive immediately; a few forward transmissions may occur before a received symbols is seen by the encoder. This dissertation explores how sparse the feedback can be while still guaranteeing performance above the rate bounds developed for sequential, non-sparse feedback communication. This dissertation introduces both a "look-ahead algorithm" and a "sub-block combining algorithm" to accomplish communication with sparse feedback.

The "look-ahead algorithm," is a sparse encoder that seeks to encode several consecutive "look-ahead" symbols without additional feedback by enforcing the same constraint developed for the sequential, non-sparse case. The number of look-ahead transmissions varies and is computed after each feedback transmission using only the feedback symbols that are shared by encoder and decoder. Simulation results show that the average number of "look-ahead" transmissions increases with message size, from slightly above one, for a 10-bit message, up to about five for an 80-bit message, for channels with capacity between 0.50 and 0.75. The "look-ahead algorithm" has significantly higher complexity than the other algorithms proposed in this dissertation.

For larger message sizes, as an alternative to the "look-ahead algorithm," the "sub-block combining algorithm" of chapter 4 can also serve as a sparse feedback encoder. In fact, the sparse version of the "sub-block combining algorithm" exhibits much lower complexity than the "look-ahead algorithm." For a given message size, the "look-ahead algorithm" achieves superior sparsity. However, the performance deficit of the "sub-block combining algorithm" decreases as the message sizes increases. The significantly lower complexity allows the "sub-block combining algorithm" to provide a practically feasible solution for much larger message sizes and for lower capacity channels than possible with the "look-ahead algorithm."

Finally, **Chapter 6** summarizes the dissertation results and conclusions, and points to possible future research direction on feedback communication and sparse feedback models.

CHAPTER 2

Algorithms, Implementation and Simulations

This chapter considers algorithms that implement sequential communication over the binary symmetric channel with noiseless feedback. The implementation framework presented here is based on the posterior patching [SF11] encoding algorithm proposed by Naghshvar *et al.* [NJW15], with their original encoding rule, as well as other rules. An implementation of this algorithm is a valuable tool that serves to validate theoretical results with simulation results, for the encoding methods that are compatible with the algorithm. Specifically, validating the achievability bounds proposed by Yang *et al.* [YPA21] was a driving motivation for the implementation of an efficient simulation framework. However, the algorithm by Naghshvar *et al.* requires that the encoder and decoder maintain and update the posterior probabilities of each candidate message in the entire message space, after every transmission. A straightforward implementation that tracks the state of each candidate message individually is prohibitively complex, since the number of messages grows exponentially with the message length. This dissertation provides an implementation framework that greatly simplifies the implementation of the binary version of Naghshvar's algorithm. This framework exploits key properties of the system to reduce the complexity of each transmission from exponential to linear, respect to the block size. The simulation results that have been obtained using this framework showcase how tight, or loose, analytical achievability bounds are respect to the encoding methods that the algorithms implement. The wide gap between previous performance bounds and the simulation results motivate the new analysis and tighter analytical bounds that are provided here and in previous works, especially [YPA21] and [AGW23]. This dissertation proposes new encoding rules for Nahshvar's transmission

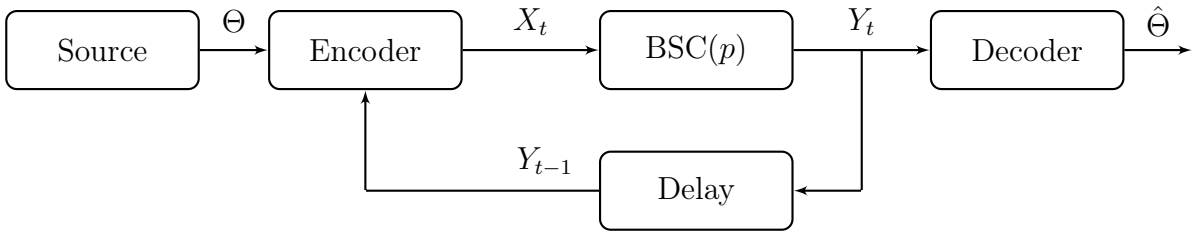

Figure 2.1: System model for the implementation of sequential communication over the BSC with a noiseless feedback channel. The information source generates a K-bit message Θ; the encoder receives the message Θ and encodes, at each time t, a binary symbols X_t that is transmitted over the forward channel. The forward channel is the binary symmetric channel (BSC), that produces binary output symbols Y_t, $t = 1, 2, \ldots$, which are. Each symbol Y_t is a noisy versions of X_t, according to the channel transition function: $\Pr(Y_t \neq X_t) = p$, where p is the channel's crossover probability. Each output symbol Y_t is sent back to the encoder through the noiseless feedback channel, and may be used to encode the next symbol X_{t+1}. At the end of the transmission, the decoder produces an estimate $\hat{\Theta}$ the message Θ, from received symbols.

scheme that greatly reduce the runtime and memory complexity and allow to obtain simulation results over the entire region of interest, where variable length feedback codes present an advantage over fixed length feedback codes, or fixed length forward error correction codes. In chapter 3, this dissertation proves that the new encoding rules satisfy the constraints needed to achieve the same achievability bounds, developed by Yang *et al.* [YPA21], for the more complex SED encoder of Naghshvar *et al.* [NJW15].

The system consists of an information source, an encoder, a noisy forward channel, a noiseless feedback channel, and a decoder, see Fig. 2.1. The algorithms proposed in this chapter implement sequential transmission. The transmission happens at discrete times index by $t = 1, 2, \ldots$. In sequential transmission the encoder sends a binary symbol $X_t \in \{0, 1\}$ to the decoder over the forward BSC channel, and the received symbol $Y_t \in \{0, 1\}$ is sent back to the encoder over the noiseless feedback channel before the next symbol is encoded. Thus, the encoder has access to every symbol received by the decoder, which it can use to encode the next symbol. In this chapter the information sequence, or message, from the source is assumed to be available to the encoder, in its entirety, before the start of

the transmission.

The problem addressed by the algorithms consists of communicating a K-bit information sequence Θ, from the encoder to the decoder. The message Θ is provided from an information source, and is assumed to be a uniform random variable, that is $\Theta \sim \mathcal{U}(\Omega)$, where $\Omega = \{0,1\}^K$. The encoder computes each next transmitted symbol X_{t+1} using the source information sequence Θ, and the feedback sequence $Y^t = Y_1, Y_2, \ldots, Y_t$. The process terminates at the earliest time $t = \tau$, when the decoder determines an estimate $\hat{\Theta}$ of Θ, such that the error probability $\Pr(\hat{\Theta} \neq \Theta) \leq \epsilon$. Where ϵ is a small threshold provided to the decoder. That is:

$$\tau \triangleq \min_{t \in \mathbb{N}} \{\exists i \in \Omega : \rho_i(y^t) \geq 1 - \epsilon\} \tag{2.1}$$

Since the encoder possesses all the information available to the decoder, it can also determine when the decoder stops the process and produces the estimate $\hat{\Theta}$. At the core of the problem is the encoding rule used by the encoder to compute each symbol X_t, to minimize the expected decoding time $\mathsf{E}[\tau]$. The purpose of the algorithms is to efficiently implement the provided encoding rules, in order to obtain simulation results and validate the achievability bounds developed for those encoding rules.

The communication schemes in this dissertation are base on the single phase "posterior matching" scheme proposed by Naghshvar *et al.* [NJW15]. A summary of the binary version of the algorithms follows. Before each transmission, both the transmitter and the receiver partition the message set $\Omega = \{0,1\}^K$ into two sets, S_0 and S_1. The partition is based on the received symbols Y^t according to a specified deterministic algorithm known to both the transmitter and receiver. Then, the transmitter encodes $X_t = 0$ if $\theta \in S_0$ and $X_t = 1$ if $\theta \in S_1$, i.e.

$$X_t = \mathrm{enc}(\theta, Y^t) = \mathbb{1}_{i \in S_1} \tag{2.2}$$

Let the posterior probabilities $\rho_i(y^t)$ of each candidate message $i \in \Omega$ be defined by:

$$\rho_i(y^t) \triangleq P(\theta = i \mid Y^t = y^t),\ \forall i \in \{0,1\}^K\,. \tag{2.3}$$

After receiving the next symbol Y_{t+1}, the receiver computes the posteriors $\rho_i(y^{t+1})$ for each message $i \in \{0,1\}^K$, via the equation:

$$\rho_i(y^{t+1}) = \Pr(\theta = i \mid Y^{t+1} = y^{t+1}) = \frac{\Pr(\theta = i, Y_{t+1} = y_{t+1} \mid Y^t = y^t)}{\Pr(Y_{t+1} = y_{t+1} \mid Y^t = y^t)}\,. \tag{2.4}$$

Using the chain rule, the probability $\Pr(\theta = i, Y_{t+1} = y_{t+1} \mid Y^t = y^t)$ in the numerator is factored into $\Pr(Y_{t+1} = y_{t+1} \mid Y^t = y^t, \theta = i)$ and $\Pr(\theta = i \mid Y^t = y^t)$. Since $\{Y^t = y^t\}$ fully determines the partitions, the first probability is q if $i \in S_{y_{t+1}}$, the set indexed by the received bit, and p if $i \notin S_{y_{t+1}}$, when i is in the other set The second factor $\Pr(\theta = i \mid Y^t = y^t)$ is just $\rho_i(y^t)$. The probability in the denominator is the marginal of the probability in the numerator, and is obtained by adding the probabilities $\Pr(Y_{t+1} = y_{t+1} \mid Y^t = y^t, \theta = j)$ for each $j \in \Omega$ as follows:

$$\rho_i(y^{t+1}) = \frac{\Pr(Y_{t+1} = y_{t+1} \mid \theta = i)\rho_i(y^t)}{\sum_{j\in\Omega} \Pr(Y_{t+1} = y_{t+1} \mid Y^t = y^t, \theta = j)\rho_j(y^t)} \tag{2.5}$$

$$= \rho_i(y^t)\frac{q\mathbb{1}_{i\in S_{y_{t+1}}} + p\mathbb{1}_{i\in S_{1\oplus y_{t+1}}}}{q\sum_{j\in S_{y_{t+1}}} \rho_j(y^t) + p\sum_{j\in\Omega\setminus S_{y_{t+1}}} \rho_j(y^t)}\,. \tag{2.6}$$

Note that $\Pr(Y_{t+1} = y_{t+1} \mid Y^t = y^t, \theta = j)$ is the same for all $j \in S_0$ and for all $j \in S_1$. It suffices to compute $P_0 \triangleq \sum_{j\in S_0} \rho_j(y^t)$, $P_1 \triangleq \sum_{j\in S_1} \rho_j(y^t)$ once. Then, the bottom term is given by $\Pr(Y_{t+1} = y_{t+1} \mid Y^t = y^t) = qP_0 + pP_1$ if $Y_{t+1} = 0$, and by $\Pr(Y_{t+1} = y_{t+1} \mid Y^t = y^t) = pP_0 + qP_1$ if $Y_{t+1} = 1$. The new posteriors $\rho_i(y^{t+1})$ can be computed from the previous

posteriors $\rho_i(y^t)$ via the following update:

$$Y_{t+1} = 0 \implies \begin{cases} \rho_i(y^{t+1}) = \rho_i(y^t)\frac{q}{qP_0+pP_1}, \forall i \in S_0 \\ \rho_i(y^{t+1}) = \rho_i(y^t)\frac{p}{qP_0+pP_1}, \forall i \in S_1 \end{cases} \tag{2.7}$$

$$Y_{t+1} = 1 \implies \begin{cases} \rho_i(y^{t+1}) = \rho_i(y^t)\frac{p}{qP_1+pP_0}, \forall i \in S_0 \\ \rho_i(y^{t+1}) = \rho_i(y^t)\frac{q}{qP_1+pP_0}, \forall i \in S_1 \end{cases} \tag{2.8}$$

The transmitter computes the same posteriors, as it has access to the received symbol Y_{t+1} via the noiseless feedback channel, which allows both transmitter and receiver to use the same deterministic partitioning algorithm. The process repeats until the first time τ that a single message i attains a posterior $\rho_i(y^\tau) \geq 1 - \epsilon$. The receiver chooses this message i as the estimate $\hat{\theta}$. Since θ is uniformly sampled, every possible message $j \in \{0,1\}^K$ has the same prior: $\Pr(\theta = j) = 2^{-K}$.

Naghshvar *et al.* proposed two methods to construct the partitions S_0 and S_1. The simplest one, which was later described as the "small enough difference" (SED) rule by [YPA21], consists of an algorithm that terminates when the SED encoding rule is satisfied:

$$\text{SED rule: } 0 \leq \sum_{i \in S_0} \rho_i(y^t) - \sum_{i \in S_1} \rho_i(y^t) < \min_{i \in S_0} \rho_i(y^t)\,. \tag{2.9}$$

The algorithm starts with an enumeration $\{1, \ldots, 2^K\}$ of all 2^K candidate messages and their posteriors $[\rho_1(y^t), \ldots, \rho_{2^K}(y^t)]$. All messages are initially in S_0, and are moved to S_1 one by one, from the smallest to the largest posterior. The partitioning process ends at any point where the SED rule (2.9) is met If the accumulated probability in S_0 falls below $\frac{1}{2}$, then the labels of S_0 and S_1 are swapped, and the process resumes.

The worst case scenario complexity of this algorithm is of order $O(M^2)$, where $M = 2^K$ is the number of posteriors. The M is squared because part of the process repeats after every swap, and in the worst case scenario the number of swaps is proportional to M. However, a

likely scenario is that the process ends after very few swaps, in which case the complexity is of order $O(M) = O(2^K)$.

The second method by which Naghshvar *et al.* proposed to construct S_0 and S_1 consists of an exhaustive search over all possible partitions, i.e., the power set 2^Ω, and use the extrinsic Jensen-Shannon divergence (EJS) metric to determine the optimal partition. This partitioning method would at least exhibit the same performance achieved by the SED rule, since the exhaustive search includes every possible partitioning, specifically, it includes that constructed with the previous algorithm.

2.1 Organization

The rest of the chapter proceeds as follows: Sec. 2.2 introduces the SEAD encoding constraint that relaxes the SED rule (2.9) criteria to significantly reduce the encoder's complexity. This rule will be analyzed further in chapter 3 to show that it satisfies the same rate bound that the SED rule does. Sec 2.3 proposes the thresholding of order posteriors (TOP) partitioning, a very simple method to construct partitions S_0 and S_1 that satisfy the SEAD rule. The TOP algorithm facilitates further complexity reduction by avoiding explicit computation of posterior updates for the majority of messages, since those posterior updates are not needed to find the threshold position. Sec. 2.4 introduces the Systematic Posterior Matching algorithm (SPM), an efficient implementation of a systematic version of the scheme by Naghshvar *et al*, that allows partitioning with either the SED or SEAS partitioning rules. The SPM algorithm is intended as a simulation framework to validate analytical results, for which a thorough description is provided, including implementation details and algorithms. Sec. 2.5 analyzes the complexity of the SPM algorithm with TOP partitioning. Sec. 2.6 provides simulation results that highlight the need for tighter performance bounds and serve to validate previously developed theoretical bounds, as well as the ones provided in Chapter 3 of this dissertation. Sec. 3.16 also provides empirical complexity results for the SPM algorithm

that validates the complexity analysis. Sec. 2.7 concludes the chapter.

2.2 The SEAD Partitioning Rule

The SED encoding rule by Naghshvar *et al.* is straightforward to implement. The algorithm could be implemented with a simple modification that results in the exact same partitions, but significantly reduces the implementation complexity. Instead of moving small items from S_0 to S_1, large items could be moved from S_1 to S_0. However, the SED rule is not without disadvantages. The implementation entails significant complexity, that grows rapidly with message sizes, even after taking additional steps to avoid many unnecessary operations. Under uniform prior, binomial prior, and some other distributions of the source information sequence, a few posterior probability values are shared by many messages at the start of the process. The implementation framework in this dissertation, also in [AGW23], and in [AYW20], relies on this posterior sharing property to greatly reduce the runtime and memory complexity. All the messages sharing an initial posterior are collected in a group. To track the posterior probabilities of all messages, it suffices to track a single posterior for each group, and a map of the messages in each group. The SED encoder would divide a group into two every time the termination condition is checked, except for the case where the condition is checked at the boundary of two groups. This holds every time the labels are swapped, and could result in multiple new groups created before every transmission. In addition, the SED partitioning rule is likely to involve messages i with very low posterior $\rho_i(y^t)$. When low posterior messages are not involved in the partitioning, many operations can be avoided, which will be described in subsequent sections.

In light of these observations, a relaxed encoding rule, the "small enough absolute difference" (SEAD) rule is proposed [AGW23]. The SEAD rule terminates when the difference between the posteriors in S_0 and S_1 is smaller than the smallest posterior in S_0, regardless whether the accumulated posterior in S_1 is more than that of S_0, if no item has a posterior

above $\frac{1}{2}$. The SEAD rule is compactly described by:

$$\left|\sum_{i\in S_0}\rho_i(y^t) - \sum_{i\in S_1}\rho_i(y^t)\right| \leq \min_{i\in S_0}\rho_i(y^t) \tag{2.10}$$

$$\rho_i(y^t) \geq \frac{1}{2} \implies S_0 = \{i\} \text{ or } S_1 = \{i\}\,. \tag{2.11}$$

The SEAD rule is much simpler to implement than the SED rule, and does not require swapping labels. When messages with shared posteriors are consolidated in groups, it is possible to implement the SEAD rule without creating more than one new group at each transmission. This dissertation shows a method, the "thresholding of ordered posteriors" [AGW23] that enforces the SEAD rule, and creates at most one new group before every transmission.

2.3 Partitioning by Thresholding of Ordered Posteriors (TOP)

The TOP rule is a simple method to construct S_0 and S_1 at any time t from the vector of posteriors $\boldsymbol{\rho}_t$, which enforces the SEAD partitioning constraint of Thm. 3. The rule requires an ordering $\{o_1, \ldots, o_M\}$ of the vector of posteriors such that $\rho_{o_1}(t) \geq \rho_{o_2}(t) \geq \cdots \geq \rho_{o_M}(t)$. TOP builds S_0 and S_1 by finding a threshold m to split $\{o_1, \ldots, o_M\}$ into two contiguous segments: $\{o_1, \ldots, o_m\} = S_0$, and $\{o_{m+1}, \ldots, o_M\} = S_1$. To determine the threshold position, the rule first determines an index $m' \in \{1, \ldots, M\}$ such that:

$$\sum_{i=1}^{m'-1} \rho_{o_i}(y^t) < \frac{1}{2} \leq \sum_{i=1}^{m'} \rho_{o_i}(y^t)\,, \tag{2.12}$$

Once m' is found, the rule must select between two possible alternatives: Either $m = m'$ or $m = m' - 1$. In other words, all that remains to decide is whether to place $o_{m'}$ in S_0 or in S_1. The selected choice is the one that guarantees that the absolute difference between P_0 and P_1 is no larger than the posterior of $o_{m'}$. Thus, the threshold m is selected from $\{m' - 1, m'\}$

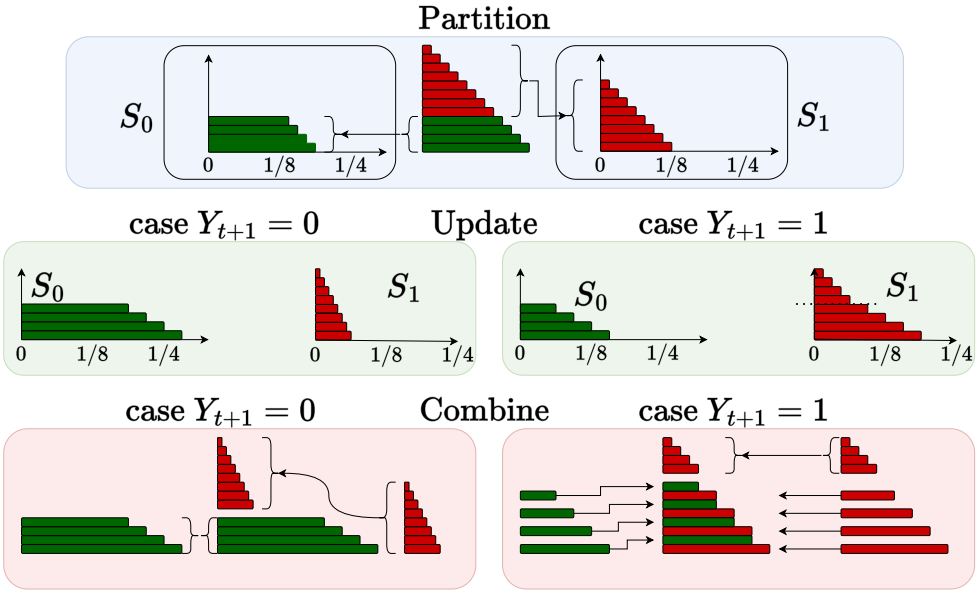

Figure 2.2: Set partitioning (top), posterior updates (center), and partition merge (bottom), for sequential transmission with the TOP partitioning rule, at each time t. The bars represent the set of messages. The bar lengths represent the message posteriors, and the bars are ordered according to posteriors (bar lengths). With TOP partition, a threshold separates the ordered list of messages (top center), into two parts, top left and top right. Updating the posteriors consists of boosting S_0 and attenuating S_1, if $Y_t = 0$ (center left), or boosting S_1 and attenuating S_0, if $Y_t = 1$ (center right). If S_0 is boosted, merging the partitions only requires to "stack" S_0 and S_1 (bottom left). If $Y_t = 1$, the items in S_0 and S_1 are interleaved to create a new sorted list (bottom right). There will be a group of items in S_1 that remain small, even after the boost. These items are "stacked" at the top the list.

as follows:

$$m = \begin{cases} m' - 1 & \text{if } \sum_{i=1}^{m'} \rho_{o_i}(t) - \frac{1}{2} > \frac{1}{2}\rho_{o_{m'}}(t) \\ m' & \text{if } \sum_{i=1}^{m'} \rho_{o_i}(t) - \frac{1}{2} \leq \frac{1}{2}\rho_{o_{m'}}(t) \end{cases} \tag{2.13}$$

Note that since $m \in \{m' - 1, m'\}$, then the posterior of $o_{m'}$ is no larger than that of o_m, and the posterior of o_m is also the value of $\rho_{\min} = \min_{i \in S_0}\{\rho_i(t)\}$.

Claim 1. *The TOP rule guarantees that the SEAD constraints of Thm. 3, given by* (3.33) *and* (2.9)*, are satisfied.*

Proof. The TOP partitioning rule sets the threshold that separates S_0 and S_1 exactly before

or exactly after item $o_{m'}$ depending on which of the cases in (2.13) occurs. To show that the TOP rule guarantees that the SEAD constraints in Thm. 3 are satisfied note that if the first case of (2.13) occurs, the threshold lies before item $o_{m'}$. Then, the inequalities in (2.12) demand that:

$$P_0 = \sum_{i=1}^{m'-1} \rho_{o_i}(y^t) = \sum_{i=1}^{m'} \rho_{o_i}(y^t) - \rho_{o_{m'}}(t) < \frac{1}{2} \tag{2.14}$$

$$P_0 > \frac{1}{2} + \frac{1}{2}\rho_{o_{m'}}(t) - \rho_{o_{m'}}(t) = \frac{1}{2} - \frac{1}{2}\rho_{o_{m'}}(t)\,. \tag{2.15}$$

When the second case of (2.13) occurs, the threshold is set after item $o_{m'}$. Then, the inequalities in (2.12) demand that:

$$P_0 = \sum_{i=1}^{m'} \rho_{o_i}(y^t) \geq \frac{1}{2} \tag{2.16}$$

$$P_0 \leq \frac{1}{2} + \frac{1}{2}\rho_{o_{m'}}(t)\,, \tag{2.17}$$

In either case:

$$\frac{1}{2} - \frac{1}{2}\rho_{o_m}(t) \leq P_0 \leq \frac{1}{2} + \frac{1}{2}\rho_{o_m}(t) \tag{2.18}$$

By definition, $\Delta = P_0 - P_1 = P_0 - (1 - P_0) = 2P_0 - 1$. Scale equation (2.18) by 2 and subtract 1, then: $-\rho_{o_{m'}}(t) < 2P_0 - 1 \leq \rho_{o_{m'}}(t)$. Then, $\mid \Delta \mid \leq \rho_{o_{m'}}(t) \leq \rho_{o_m}(t) = \rho_{\min}$. This concludes the proof. □

The TOP rule shows that the construction of partitions S_0 and S_1 that satisfy the SEAD rule could be done in three simple steps. The first is finding the threshold item $o_{m'}$ where the c.d.f. induced by the ordered vector of posteriors crosses $\frac{1}{2}$. The second is deciding if the threshold should be placed before or after item $o_{m'}$. And the third is allocating all items before the threshold to S_0 and all items after the threshold to S_1.

2.4 The Systematic Posterior Matching Algorithm

The SPM-TOP algorithm is a low complexity scheme that implements sequential transmission over the BSC with noiseless feedback with a source message sampled from a uniform distribution. Algorithms 3 and 4 present the encoder and decoder as functions that, when called, respectively transmit and decode one channel symbol. These functions have the usual communication and confirmation phases, but the communication phase starts with a systematic transmission phase. Thus systematic transmissions of the communication phase are treated as a separate systematic phase, for a total of three phases that we proceed to describe in detail.

2.4.1 Systematic Phase

Let the sampled message be $\theta \in \{0,1\}^K$, with bits $b_i^{(\theta)}$, that is $\theta = \{b_1^{(\theta)}, b_2^{(\theta)}, \ldots, b_K^{(\theta)}\}$. For $t = 1, \ldots, K$, the bits $b_t^{(\theta)}$ are transmitted (See line 2 of Alg. 3.) and the vector $y^K \triangleq \{y_1, \ldots, y_K\}$ is received without any decoder computations. (See line 2 of Alg. 4.)

At time $t = K$, right after the K-th systematic transmission, both transmitter and receiver initialize a list of $K+1$ groups $\{\mathcal{G}_0, \ldots, \mathcal{G}_K\}$, where each $\mathcal{G}_i$ is a tuple $\mathcal{G}_i = [N_i, L_i, h_i, \rho_i(y^t)]$. Alg. 1 implements the initialization of the list of groups as the function **InitializeGroups**, which is called in line 5 of Alg. 3 and Alg. 4.

As described in Alg. 1 for Group i, N_i is the count of messages in the group; L_i is the index of the first message in the group; h_i is the shared Hamming distance between y^K and any message in the group, that is: $l, s \in \mathcal{G}_i \implies \sum_{j=1}^K b_j^{(l)} \oplus y_j = \sum_{j=1}^K b_j^{(s)} \oplus y_j = h_i$; and $\rho_i(y^t)$ is the posterior shared by every item in the group. At time $t = K$, each group $\mathcal{G}_i, i = 1, \ldots, K$ has that $N_i = \binom{K}{i}$, $L_i = 0$, $h_i = i$, and $\rho_i(K) = p^j q^{K-j}$. The groups are initially established in order of decreasing probability, equivalent to increasing Hamming weight, since for $p < q$, $j > l \implies p^l q^{K-l} < p^j q^{K-j}$, (see line 2 of Alg. 1).

At the end of the systematic phase, the transmitter finds the index h_θ of the group, and

Algorithm 1: $\mathbf{G} = \textbf{InitializeGroups}(p, q, K)$

Input: Channel parameters p, q, message length K
Output: List of Groups $\mathbf{G} = \mathcal{G}_0, \ldots, \mathcal{G}_K$
for i=0,...,K **do**
  $\rho_i \leftarrow q^{K-i}p^i$ ▷ $\forall j \in \mathcal{G}_i \rightarrow \rho_j(K) = \rho_i$
  $N_i \leftarrow \binom{K}{i}$ ▷ $N_i \triangleq \mid \mathcal{G}_i \mid$: count of items in group
  $L_i \leftarrow 0$ ▷ First index in group
  $h_i \leftarrow i$ ▷ Shared group Hamming weight
  $W_i = 1$ ▷ Delayed update coefficient
  $f_i = 0$ ▷ For encoder: set $f_i = 1$ if $\theta \in \mathcal{G}_i$
  $\mathcal{G}_i = (N_i, L_i, h_i, \rho_i, W_i, f_i)$
end
$\mathbf{G} \leftarrow \mathcal{G}_0, \ldots, \mathcal{G}_K$

the index n_θ within the group, that corresponds to the sampled message θ, using Alg. 2. Line 5 of Alg. 3 shows the call to the function **InitializeGroups** of Alg. 2. The group index h_θ is given by $h_\theta = \sum_{j=1}^{K} o_j^{(\theta)} \oplus y_j$ and the index $n_\theta \in \{0, \ldots, \binom{K}{h_\theta} - 1\}$ within the group is computed using the function **FindMessageIndex** of Alg. 2

Algorithm 2: (h, n)=$\textbf{FindMessageIndex}(K, x^K, y^K)$

Input: Length K, Trans. Seq. x^K, Rec. Seq. y^K
Output: Hamming weight h, index n
$e^K = x^K \oplus y^K$
$h \leftarrow \sum_{j=0}^{K} e_j^i$
$n \leftarrow 0$
$c \leftarrow h$
for $j = 0, \ldots, K-1$ **do**
  if $c = 0$ **then**
    Break
  else if $e_j = 0$ **then**
    $n \leftarrow n + \binom{K-j-1}{c-1}$
  else
    $c \leftarrow c - 1$
  end
end

2.4.2 Communication Phase

The communication phase consists of the transmissions after the systematic phase, for which all the posteriors are below the threshold $\frac{1}{2}$. During communication phase, the transmitter attempts to boost the posterior of the transmitted message, past the threshold $\frac{1}{2}$, though any other message may cross the threshold instead, due to channel errors.

The list of groups initialized in the systematic phase is maintained ordered by decreasing common posterior. The list of groups is partitioned into S_0 and S_1 before each transmission using rule (2.12) as shown in Alg. 6. For this, the group $\mathcal{G}_{n_m}$ that contains the threshold item o_m is found first, see lines $4-7$ of Alg. 6. Then all groups before $\mathcal{G}_{n_m}$ are assigned to S_0 and all the groups after $\mathcal{G}_{n_m}$ are assigned to S_1, shown in line 8 of Alg. 6. The group $\mathcal{G}_{n_m}$ is split into two groups, one of which is assigned to S_0 and the other to S_1. Note that the item o_m, that sets the threshold, is a member of $\mathcal{G}_{n_m}$. To determine how the group $\mathcal{G}_{n_m}$ is split, the index n_m of item o_m, within group $\mathcal{G}_{n_m}$, needs to be computed first, see lines $9-12$ of Alg. 6. The TOP rule demands that all items $j \in \mathcal{G}_{n_m}$ with index $n_m^{(j)} \leq n_m$ be assigned to S_0 and all items $i \in \mathcal{G}_{n_m}$ with index $n_m^{(i)} > n_m$ to S_1. Group $\mathcal{G}_{n_m}$ is split into two by creating an new group with the segment of items past n_m that belongs in S_1, as shown in lines $12-27$ of Alg. 6. However, if the item with index n_m is the last item in $\mathcal{G}_{n_m}$, the entire group $\mathcal{G}_{n_m}$ belongs in S_0. Splitting the group is avoided in this particular case, as shown in lines $28-34$ of Alg. 6. This case is more likely to happen to the end of the communication phase, when the first few groups have low cardinality and high total posterior. The same partitioning algorithm is used by the encoder and decoder during the communication phase, as shown in the calls to function **PartitionGroups** in line 31 of Alg. 3, and in line 23 of Alg. 4.

After each transmission t, the posterior probabilities of the groups are updated using the received symbol Y_t according to equation (3.84), which is shown by Alg. 5. Each posterior is multiplied by a weight update, computed using equation (3.85), according to its assignment,

S_0 or S_1, see lines 4, 8, and 15 of Alg. 5. Then, the lists that comprise S_0 and S_1 are merged into a single sorted list, see lines $11-12$, $16-17$, and 21 of Alg. 5. The process repeats for the next transmission, which is shown by the calls to functions **CommPhaseUpdate** in lines 23 and 27 of Alg. 3 and lines 15 and 19 of Alg. 4. The communication phase is interrupted to transition to the confirmation phase when the posterior of a candidate message crosses the $\frac{1}{2}$ threshold. The communication phase might resume if the posterior of message i that triggered the confirmation falls below $\frac{1}{2}$ rather than increasing past $1-\epsilon$, and all other posteriors are still below $\frac{1}{2}$, see line 22 of Alg. 3, and line 14 of Alg. 4.

2.4.3 Confirmation Phase

The Confirmation Phase is triggered when a candidate i attains a posterior $\rho_i(y^t) \geq \frac{1}{2}$, see line 33 of Alg. 3 and line 24 of Alg. 4. During this phase the transmitter will attempt to boost $\rho_i(y^t)$, the posterior of candidate i, past the $1-\epsilon$ threshold, if it is the true message θ. Otherwise it will attempt to drive its posterior below $\frac{1}{2}$. Clearly, the randomness of the channel could allow the posterior $\rho_i(y^t)$ to grow past $1-\epsilon$, even if it is the wrong message, resulting in a decoding error. Alternatively, the right message could still fall back to the communication phase, also due to channel errors. Let $U_i(t)$ denote the log likelihood ratio defined by:

$$U_i(t) \triangleq \log_2\left(\frac{\rho_i(Y^t)}{1-\rho_i(Y^t)}\right). \tag{2.19}$$

The confirmation phase lasts for as long as the posterior of the message that triggered its start stays between $\frac{1}{2}$ and $1-\epsilon$ or equivalently $U_i(t)$ stays between 0 and $\epsilon_U \triangleq \log_2\left(\frac{1-\epsilon}{\epsilon}\right)$.

There are no partitioning, update, or combining operations during the confirmation phase. If j is the message in the confirmation phase, then the partitioning is just $S_0 = \{j\}$, $S_1 = \Omega \setminus \{j\}$. A single update is executed, if a fall back to to the communication phase occurs, letting $\rho_i(y^t) = \rho_i(y^{T_n}) \quad \forall i = 1, \ldots, M$, where n is the index of the confirmation phase

round that just ended, and T_n is the time at which it started, see the calls to **CommPhase-Update** in line 23 of Alg. 3, and line 14 of Alg. 4. This is because every negative update that follows a positive update results in every $\rho_i(y^t)$ returning to the state it was at time $t-2$. This is summarized in claim 3 that follows. During the confirmation phase it suffices to check if $U_j(t) \geq \epsilon_U$, in which case the process should terminate, or if $U_j(t) < 0$, in which case a fall back occurs. The encoder terminates the process by computing an estimate $\hat{x}^K$ of the systematic symbols vial Alg. 7, which is shown in the call to function **IndexToEstimate** in line 9 of Alg. 4. When the process terminates, the transmitter can declare a successful decoding if the index n_0 and Hamming distance h_0 of the message in confirmation match n_θ and h_θ, and an error otherwise, as shown in lines $12-16$ of Alg. 3.

During the confirmation phase we only need to count the difference between boosting updates and attenuating updates. Since the $U_i(t)$ changes in steps with magnitude C_2, then there is a unique number N such that $U_i(T_n) + NC_2 \geq \epsilon_U$ and $U_i(T_n) + (N-1)C_2 < \epsilon_U$. Starting at time $t = T_n$, since $S_0 = \{j\}$, any event $Y_{t+1} = 0$ is a boosting update that results in $U_i(t+1) = U_i(t) + C_2$ and any event $Y_{t+1} = 1$ is an attenuating update that results in $U_i(t+1) = U_i(t) - C_2$. A net of N boosting updates are needed to reach $U_i(T_n) + NC_2$. Let the difference between boosting and attenuating updates be $Z(t) \triangleq \sum_{s=T_n+1}^{t}(1-2Y_s)$. The transmission terminates the first time τ where $Z(\tau) = N$. However, a fall back occurs if $Z(t)$ ever reaches -1 before reaching N. The value of N can be computed as follows: let $N_1 \triangleq \left\lceil C_2^{-1} \log_2\left(\frac{1-\epsilon}{\epsilon}\right)\right\rceil$, and let $\epsilon_n \triangleq \log_2\left(\frac{1-\epsilon}{\epsilon}\right) - N_1 C_2$, see line 35 of Alg. 3, and line 26 of Alg. 4. Suppose the confirmation phase starts at some time $t = T_n$, then, $N = N_1$ if $U_i(T_n) \geq \epsilon_N$, otherwise $N = N_1 + 1$. Once N is computed, all that remains is to track $Z(t)$, where $Z(t+1) = Z(t) + (1 - 2Y_{t+1})$, and return to the communication phase if $Z(t)$ reaches -1, or terminate the process if $Z(t)$ reaches N, see lines 18 and 22 of Alg. 3, and lines 11 and 14 of Alg. 4.

Algorithm 3: $(S_0', S_1', Z', N_\epsilon')$=**Encoder**$(t, y^t, S_0, S_1, Z, N_\epsilon)$

Data: Channel constants p, q, error bound ϵ, length K
Input: Message θ time t, feedback $y^t = [y_1, \ldots, y_t]$ and h_θ, n_θ ▷ set once, see line 4.
Input: Previous Encoder state: partitions S_0, S_1, confirmation state Z, target N_ϵ
Output: Next symbol x_{t+1}, encoder state S_0', S_1', confirmation state Z', target N_ϵ'
if $t < K$ **then**
  channel input: $x_{t+1} = b_{t+1}^\theta$ ▷ $\theta = [b_1^{(\theta)}, \ldots, b_K^{(\theta)}]$
else if $t = K$ **then**
  $\mathbf{G} \leftarrow$ **InitializeGroups**(K, p, q) ▷ Alg. 1
  $(h_\theta, n_\theta) \leftarrow$ **FindMessageIndex**(K, x^K, y^K) ▷ Alg. 2
  $m_\theta \leftarrow h_\theta$, $f_{m_\theta} \leftarrow 1$ ▷ Since $\theta \in \mathcal{G}_{h_\theta}$, then $m_\theta \leftarrow h_\theta$ and $f_{h_\theta} = 1$ to track $\mathcal{G}_{h_\theta}$
  $Z' \leftarrow -1$ ▷ Set communication phase
else if $Z \geq 0$ **then**
  $Z' \leftarrow (y_t = 0 \ \mathbf{?}\ Z + 1 \ \mathbf{:}\ Z - 1)$
  if $Z' = N_\epsilon$ **then**
    if $h_0 = h_\theta$ *and* $L_0 = n_\theta$ **then**
      Declare success
    else
      Declare error
    end
    $S_0', S_1' \leftarrow \emptyset$ ▷ Terminate process
  else if $Z' \geq 0$ **then**
    $S_0', S_1', \leftarrow S_0, S_1, N_\epsilon$ ▷ No change in partitions
    $N_\epsilon' \leftarrow N_\epsilon$ ▷ Same target
    $x_{t+1} \leftarrow (h_0 = h_\theta$ **and** $L_0 = n_\theta$ **?** 0 **:** $1)$
  else if $Z' = -1$ **then**
    $\mathbf{G}' \leftarrow$ **CommPhaseUpdate**(S_0, S_1, y_t) ▷ Alg. 5 and Fall to comm. phase
  end
else if $Z = -1$ **then**
  $\mathbf{G}' \leftarrow$ **CommPhaseUpdate**(S_0, S_1, y_t) ▷ Alg. 5
  $Z' \leftarrow Z$ ▷ No state change yet (see line 3)
end
if $Z' = -1$ **then**
  $S_0', S_1' \leftarrow$ **PartitionGroups**$(\mathbf{G}')$ ▷ Alg. 6
  $x_{t+1} \leftarrow (\exists \mathcal{G}_i \in S_0\text{:}\ f_i = 1 \ \mathbf{?}\ 0 \ \mathbf{:}\ 1)$
  if $\rho_0 \geq \frac{1}{2}$ ***and*** $N_0 = 1$ **then**
    $Z' \leftarrow 0$ ▷ Set Confirmation phase start
    $N_\epsilon' \leftarrow \left\lceil C_2^{-1}\left(\log_2\left(\frac{1-\epsilon}{\epsilon}\right) - \log_2\left(\frac{\rho_0}{1-\rho_0}\right)\right)\right\rceil$
  end
end

Algorithm 4: $(S_0', S_1', Z', N_\epsilon')$=**Decoder**$(t, y^t, S_0, S_1, Z, N_\epsilon)$

Data: Channel constants p, q, error bound ϵ, length K
Input: Previous Decoder state of partitions S_0, S_1
Input: Previous confirmation state Z, target N_ϵ
Output: Next Decoder state S_0', S_1'
Output: Next confirmation state Z', next target N_ϵ'
if $t < K$ **then**
  ▷ No action needed at decoder
else if $t = K$ **then**
  $\mathbf{G} \leftarrow$ **InitializeGroups**(K, p, q) ▷ Alg. 1
  $Z' \leftarrow -1$ ▷ Set communication phase
else if $Z \geq 0$ **then**
  $Z' \leftarrow (y_t = 0 \ ? \ Z + 1 \ : \ Z - 1)$
  if $Z' = N_\epsilon$ **then**
    $\hat{x}^K \leftarrow$ **IndexToEstimate**$(h_0, L_0, y_{1:K}^t)$ ▷ Alg. 7
    Report Estimate $\hat{x}^K$
  else if $Z' \geq 0$ **then**
    $S_0', S_1', \leftarrow S_0, S_1, N_\epsilon$ ▷ No change in partitions
    $N_\epsilon' \leftarrow N_\epsilon$ ▷ Same target
  else if $Z' = -1$ **then**
    $\mathbf{G}' \leftarrow$ **CommPhaseUpdate**(S_0, S_1, y_t)
    ▷ Fall back to communication phase: Alg. 5
  end
else if $Z' = -1$ **then**
  $\mathbf{G}' \leftarrow$ **CommPhaseUpdate**(S_0, S_1, y_t) ▷ Alg. 5
  $Z' \leftarrow Z$ ▷ No state change yet (see line 4)
end
if $Z' = -1$ **then**
  $S_0', S_1' \leftarrow$ **PartitionGroups**$(\mathbf{G}')$ ▷ Alg. 6
  if $\rho_0 \geq \frac{1}{2}$ *and* $N_0 = 1$ **then**
    $Z' \leftarrow 0$ ▷ Set confirmation phase start
    $N_\epsilon' \leftarrow \left\lceil C_2^{-1}\left(\log_2\left(\frac{1-\epsilon}{\epsilon}\right) - \log_2\left(\frac{\rho_0}{1-\rho_0}\right)\right)\right\rceil$
  end
end

Algorithm 5: $\mathbf{G} = \textbf{UpdateGroupList}(y_t, S_0, S_1)$

Input: Channel output: y_t, partitions S_0, S_1
Output: Updated list $\mathbf{G} = \{\mathcal{G}_0, \ldots, \mathcal{G}_{K+n_s}\}$

$w_0 \leftarrow \frac{q}{P_{y_{t+1}}(q-p)+p}$ ▷ Weight update for items in S_0
$w_1 \leftarrow \frac{p}{P_{y_{t+1}}(q-p)+p}$ ▷ Weight Update for items in S_1
$m_0 \leftarrow 0$, $m_1 \leftarrow 0$ ▷ Indices of first group in S_0, S_1
$W_{m_1} \leftarrow W_{m_1} \cdot w_1$ ▷ Update weight of first group in S_1
$\mathbf{G} \leftarrow \emptyset$ ▷ Initialize to null
while $S_0 \neq \emptyset$ **do**
  if $w_0 \cdot \rho_{m_0} < W_{m_1} \cdot \rho_{m_1}$ **then**
    $\rho_{m_1} \leftarrow \rho_{m_1} \cdot W_{m_1}$
    $W_{m_1+1} \leftarrow W_{m_1+1} \cdot W_{m_1}$ ▷ Update Next weight
    $W_{m_1} \leftarrow 1$ ▷ Reset weight W_{m_1}
    remove $\mathcal{G}_{m_1}$ from S_1
    insert $\mathcal{G}_{m_1}$ to tail of $\mathbf{G}$
    $m_1 \leftarrow m_1 + 1$ ▷ Get next item from S_1
  else
    $\rho_{m_0} \leftarrow \rho_{m_0} \cdot w_0$ ▷ Update $\rho_{m_0}(t)$
    remove $\mathcal{G}_{m_0}$ from S_0
    append $\mathcal{G}_{m_0}$ to tail of $\mathbf{G}$
    $m_0 \leftarrow m_0 + 1$ ▷ Get next item from S_0
  end
end
append S_1 to tail of $\mathbf{G}$ ▷ No update for rest of S_1

Algorithm 6: $(S_0, S_1) = \textbf{PartitionGroups}(\mathbf{G}, n_\theta)$

Input: List of Groups $\mathbf{G} = \{\mathcal{G}_0, \ldots, \mathcal{G}_{K+n_s}\}$ ▷ n_s: new group created by line 6
Input: (Transmitter only) index n_θ ▷ Encoder: tracks group with $h_i = h_\theta, n_i = n_\theta$
Output: Sets S_0, S_1 that partition $\mathbf{G}$ ▷ $S_0 = \{\mathcal{G}_0, \ldots \mathcal{G}_m\}$, $S_1 = \{\mathcal{G}_{m+1}\}, \ldots \mathcal{G}_{K+n_s}\}$

$m \leftarrow 0$ ▷ Index of first group in $\mathbf{G}$
$S_0 \leftarrow \emptyset$ ▷ Initialize S_0 to empty
$P_0 \leftarrow 0$ ▷ Posterior in S_0: $P_0 \triangleq P(\theta \in S_0)$
while $P_0 + N_m \rho_m < \frac{1}{2}$ **do**
  $P_0 \leftarrow P_0 + N_m \rho_m(t)$ ▷ $N_m, \rho_m \in \mathcal{G}_m$
  $m \leftarrow m + 1$ ▷ Increase group index m
end
$S_0 \leftarrow \{\mathcal{G}_0, \ldots, \mathcal{G}_{m-1}\}$, $S_1 \leftarrow \{\mathcal{G}_{m+1}, \ldots\}$
$n \leftarrow \lceil \frac{0.5 - P_0}{\rho_m(t)} \rceil$ ▷ Initial n value
if $P_0 + n\rho_m(t) > \frac{1}{2}(1 + \rho_m(t))$ **then**
  $n \leftarrow n - 1$ ▷ **TOP rule**
end
if $n > 0$ *and* $n < N_m$ **then**
  $N_{right} \leftarrow N_m - n$ ▷ Items in new group
  $L_{right} \leftarrow L_m + n$ ▷ Index of first item
  $N_{left} \leftarrow n$ ▷ Decrease count in old group
  ▷ Lines 18-19 only execute for transmitter.
  if $f_m = 1$ *and* $n_\theta \geq L_{right}$ **then**
    $f_{right} \leftarrow 1$ ▷ Case: $L_{right} \leq n_\theta < L_{right} + n$
    $f_{left} \leftarrow 0$ ▷ $\mathcal{G}_m$ no longer group containing n_θ
  else
    $f_{right} \leftarrow 0$ ▷ Case: $L_m \leq n_\theta < L_{right}$
  end
  $\mathcal{G}_m^{(left)} \leftarrow (N_{left}, L_m, h_m, \rho_m, W_m, f_{left})$
  $\mathcal{G}_m^{(right)} \leftarrow (N_{right}, L_{right}, h_m, \rho_m, W_m, f_{right})$
  $S_0 \leftarrow S_0 \cup \mathcal{G}_m^{(left)}$
  $S_1 \leftarrow \mathcal{G}_m^{(right)} \cup S_1$
else if $n = 0$ **then**
  $S_1 = \mathcal{G}_m \cup S_1$ ▷ Case entire $\mathcal{G}_m$ belongs in S_1
  $m \leftarrow m - 1$ ▷ Equation (2.13)
else if $n = N_m$ **then**
  $S_0 = S_0 \cup \mathcal{G}_m$ ▷ Case entire $\mathcal{G}_m$ belongs in S_0
  $P_0 \leftarrow P_0 + n\rho_m(t)$
end

Algorithm 7: $\hat{x}^K = \textbf{IndexToEstimate}(h, n, K, y^K)$

Input: Weight h, index n, length K symbols y^K
Output: Estimate $\hat{x}^K$ ▷ Estimate of $\theta = x^K$
$\hat{x}^K \leftarrow y^K$ ▷ Initialize $\hat{x}^K$ with received symbols y^K
for $j = 0, \ldots, K-1$ **do**
  if $h = 0$ **then**
    Break
  else if $n < \binom{K-1-j}{h-1}$ **then**
    $\hat{x}_j \leftarrow \neg \hat{x}_j$
    $h \leftarrow h - 1$
  else
    $n \leftarrow n - \binom{K-j-1}{h-1}$
  end
end

2.5 Complexity of the SPM-TOP Algorithm

The memory complexity of the SPM-TOP algorithm is of order $O(K^2)$ because we use a triangular array of all combinations of the form $\binom{K}{i} \quad i \in \{0, \ldots, K\}$. The algorithm itself stores a list of groups that grows linearly with K, since the list size is bounded by the decoding time τ.

The time complexity of the SPM-TOP algorithm is of order $O(K^2)$. To obtain this result note that the total number of items that the system tracks is bounded by the transmission index t. At each transmission t, the partitioning, update, and combine operations require visiting every item at most once. Furthermore, because of the complexity reduction described in Sec. 2.4.2, the system executes operations for only a fraction of all the items that are stored. The time complexity at each transmission is then of order $O(K)$, with a small constant coefficient. The number of transmissions required is approximately K/C as the scheme approaches capacity. A linear number of transmissions, each of which requires a linear number of operations, results in an overall quadratic complexity, that is, order $O(K^2)$, for fixed channel capacity C. The K systematic transmissions only require storing the bits, and in the confirmation phase we just add each symbol Y_t to the running sum. The

complexity of this phase is then of order $O(K)$. Therefore, the complexity of $O(K^2)$ is only for the communication phase.

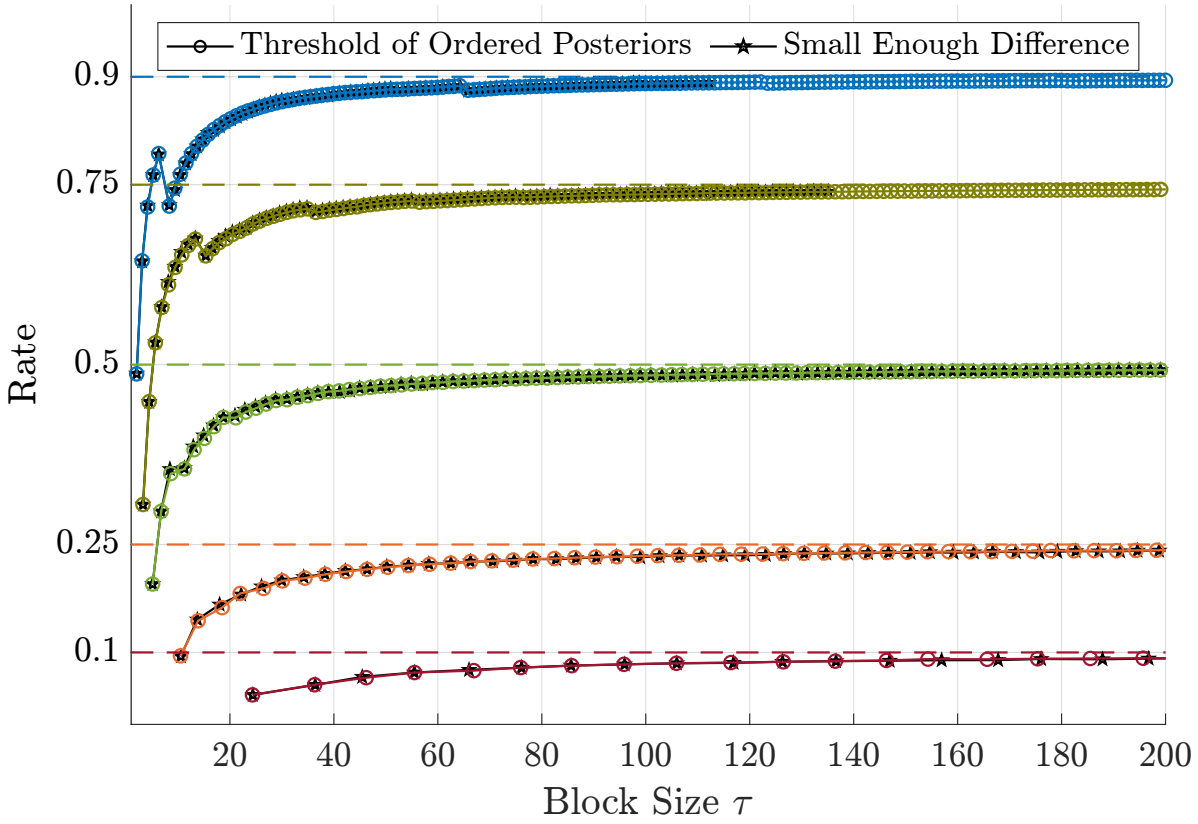

Figure 2.3: Rate performance of the SED and SEAD encoders vs. blocklength. The rate performance of the SEAD encoder, solid curves with dots $-\circ$ labeled SPM SEAD algorithm, over channels with capacities $0.90, 0.75, 0.50, 0.25, 0.10$ are provided. For comparison, The rate performance of the SED algorithm is also provided, solid curves $-\star$, labeled SPM-SED Algorithm. The channel capacities are shown for reference, see horizontal $--$ lines.

2.6 Simulation Results of the SPM Algorithm

Performance results from simulation of the systematic posterior matching algorithm are provided in Fig. 2.3 and Fig. 2.4. The rate curves show average rate $K/\mathsf{E}[\tau]$ vs. average blocklength $\mathsf{E}[\tau]$. Fig. 2.3 shows the rate performance of the SEAD encoder implemented with the TOP partitioning rule and the rate performance of the original SED encoder by

Naghshvar *et al.* [NJW15], over channels with capacity $C = 0.90, 0.75, 0.50, 0.25, 0.10$. The performance of the the SEAD encoder is shown with the solid colored curves $-\circ$ labeled "Threshold of Ordered Posteriors," and the performance of the SED encoder is shown with the black solid curves $-\star$ with stars. The simulations are for channels with with capacities $0.1, 0.25, 0.5, 0.75, 0.9$, shown by the dashed lines $--$, and the decoding threshold used was $\epsilon = 1e-3$. The simulated rate curves attain an average rate that approaches capacity rapidly. The SPM algorithm with the thresholding of ordered posterior rule exhibits a rate performance that is indistinguishable from that of the algorithm implementing the SED rule. This indistinguishable performance motivates the analysis shown in Chapter 3.

The rate curves of Fig. 2.3 and Fig. 2.4 are jagged, which is explained by a high error approaching the threshold $\epsilon = 1e-3$ at the peaks, and dropping far below the threshold at the troughs, as shown in the bottom of Fig 2.4. To better understand this behavior, Fig 2.4 includes simulations of the SEAD algorithm with a pseudo random stopping rule. The rate performance with the pseudo random rule is shown in the top of Fig 2.4 with the dash dot curves $-.\circ$ labeled "Randomized SPM-TOP," along with rate of the standard stopping rule (2.1), solid $-\circ$ colored curves. The the pseudo-random stopping rule alternates between the standard rule, which is stopping when a message i attains $U_i(t) \geq \log_2((1-\epsilon)/\epsilon)$, and an early stopping rule, when a message attains $U_i(t) \geq \log_2((1-\epsilon)/\epsilon) - C_2$, and requires one less correctly received transmission than the standard rule. A pseudo random function selects either rule with a probability biased to obtain a higher rate by forcing the FER to be close to the threshold ϵ, rather than upper bounded bounded by ϵ. Let the bias probability of early decoding be π_1, and let i be the message in confirmation, with $U_i(t) \geq \log_2((1-\epsilon)/\epsilon) - C_2$, then π_1 satisfies:

$$\pi_1 U_i(t) + (1-\pi_1)(U_i(t) + C_2) = \log_2\left(\tfrac{1-\epsilon}{\epsilon}\right) \tag{2.20}$$

$$\pi_1 = 1 - \frac{1}{C_2}\left(\log_2\left(\tfrac{1-\epsilon}{\epsilon}\right) - U_i(t)\right) \tag{2.21}$$

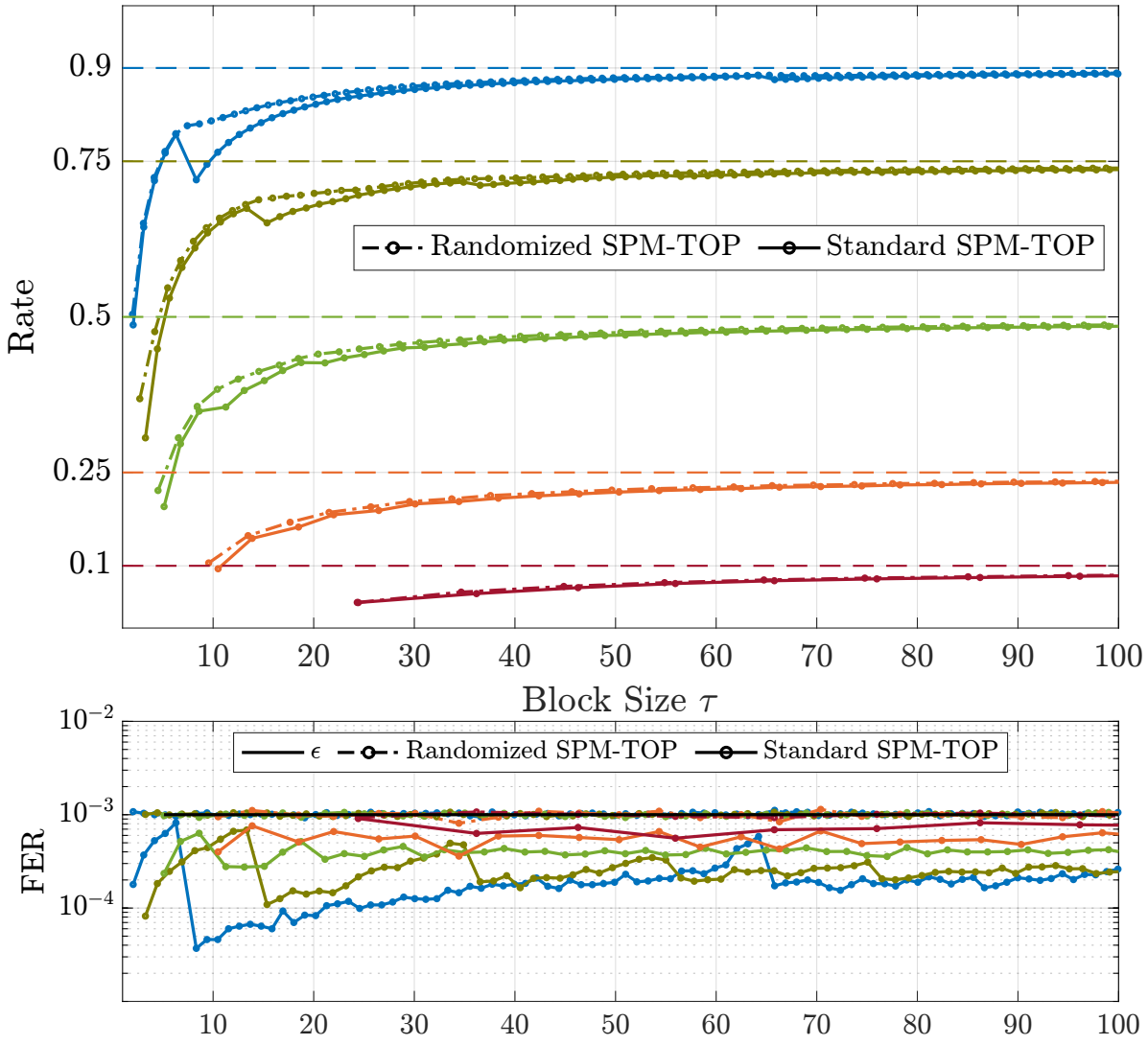

Figure 2.4: Rate performance (top) of the SPM algorithm with two stopping rules, and with thresholding of ordered posteriors (TOP) partitioning. The rate performance of SPM with a pseudo random stopping rule is shown with the dash dot $-.\bullet$ curves labeled "Randomized SPM-TOP." The pseudo random rule allows early decoding with a probability biased to attain a frame error rate (shown in the bottom) close to the threshold $\epsilon = 1e-3$, instead of the standard rule (2.1) that requires a frame error rate below the threshold. The rate curve for the standard stopping rule (2.1) is shown with the solid lines $-\bullet$. The partitioning method for all the curves is the thresholding of ordered posteriors (TOP), which implements the SEAD encoder. The frame error rate curves obtained for each of the rate curves is shown in the bottom. are shown in the bottom.

The bottom plot of Fig. 2.4 shows the frame error rate for the rate curves in the top. The FER curves that are below the horizontal line at $\epsilon = 1e-3$, and far below for at some blocklengths, are for the standard stopping rule, while the FER curves for the pseudo random stopping rule are tightly packed around the threshold $\epsilon = 1e-3$. The rate curves obtained with the pseudo random stopping rule are smooth compared to the standard rule. This, along with sudden drops in the FER of the standard rule, support the explanation that the peaks and troughs in the rate curves of the standard rule are caused by frequent events where a message i attains a high posterior, close to $\rho_i(t) \lessapprox 1-\epsilon$, but not high enough to terminate the process. Such events force at least one additional transmission, and thus the process often terminates with a posterior $\rho_i(\tau)$ high above $1-\epsilon$, or an error probability $1-\rho_i(\tau)$ far below ϵ, as shown by the left of the FER curves at the bottom of Fig. 2.4.

2.6.1 Complexity Results of the SPM Algorithm

The complexity analysis of the SPM-TOP algorithm described in 2.5 is validated with simulation results. Fig. 2.5 shows the average time per message and per 1000 symbols, in milliseconds, for the original SED encoder, and for the SEAD encoder implemented by thresholding of ordered posteriors partitioning, for a channel with capacity $C = 0.50$. The original SED encoder is implemented by the systematic posterior matching SPM algorithm that groups messages of with equal posterior to lower the complexity. However, the posterior of every group is updated after every transmission. Also, many posterior groups need alternatively assigned to S_0 and S_1 before the SED rule is satisfied for the first time. The simulated time of the this algorithm is shown by the blue and magenta curves of Fig. 2.5.

The SEAD encoder is implemented by the thresholding of ordered posteriors (TOP), which is much simpler than the SED algorithm. Partitioning with the TOP rule only requires to visit the groups with largest posteriors, until the group that contains the "weighted median," where the c.d.f. crosses half, is found. These larger groups are all assigned to the set S_0. When the TOP rule is used, the SPM algorithm performs posterior updates on the

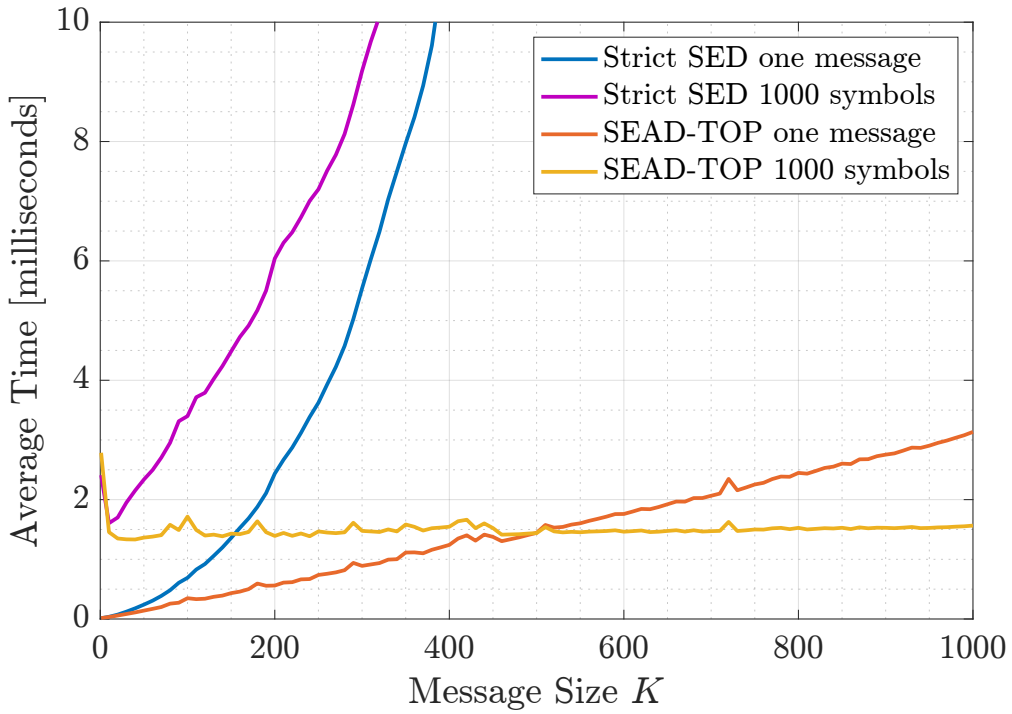

Figure 2.5: Shows average runtime of the SED and SEAD encoder, in milliseconds per message and per each 1000 symbols, as a function of message size K, for a channel with capacity $C = 0.50$. The average time per message of the SED encoder is the solid blue line; the average time per each 1000 symbols for the SED encoder is the magenta line; the average time per message of the SED encoder implemented by thresholding of ordered posteriors partitioning is the orange solid line; and the average time per 1000 symbols of the SEAD encoder is the yellow solid line. The time per 1000 symbols is provided instead of per each symbol, to show all the curves in the same plot. Note that for this channel, with $C = 0.50$, at $K = 500$ the average number of symbols required is about 1000, and is where the SEAD curves cross. The number of trials used to obtain this data is $100,000$.

all the posterior groups in S_0. From the posterior groups in S_1 only those whose updated posterior exceed the smallest posterior in S_0 are updated, which is zero when $Y_t = 0$, and a few groups when $Y_t = 1$. This results in a much simpler algorithm, as can be seen in the runtime curves of Fig. 2.5.

For a more accurate characterization of the complexity's evolution as a function of message size K, the simulations also counted the number of posterior groups involved in the set partitioning, and in the posterior updates and partition merge operations executed during

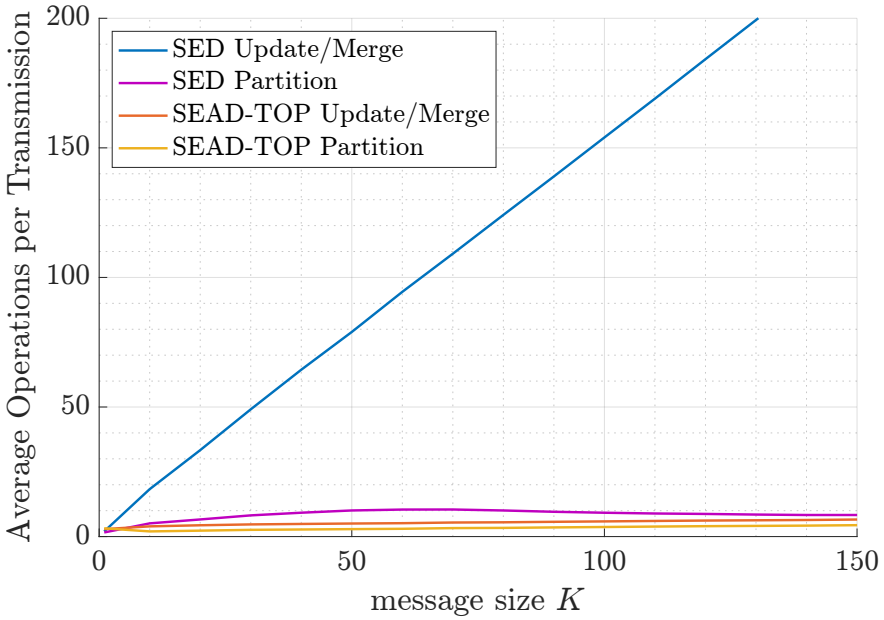

Figure 2.6: Average number of posterior groups operated on at each symbol transmission as a function of message size K, for the SPM algorithm implementing the original SED encoder and the relaxed SEAD encoder. The algorithm compares posteriors to construct the partitions S_0 and S_1 before the transmission, and then updates the posteriors and merges the partitions after the each transmission. The solid blue line labeled "SED Update/Merge" is the number of update/merge operations for the SED encoder; the solid magenta curve labeled "SED Partition" is the number of partition operations for the SED encoder; the orange and yellow curves are the operations for the SEAD encoder, which are shown at scale in Fig. 2.6. Both simulations are for a channel with capacity $C = 0.50$.

the transmission of each symbol and each message. Counting the operations prevents external factors like computer temperature and OS tasks that compete for computing resources with the simulations, from affecting the complexity results. The average number of operations per symbol are shown in Fig. 2.6 and 2.7, and the number of operations per message in 2.8 and 2.9. Fig. 2.6 shows that the number of posterior updates of the original SED encoder is a linear function of message size K, but with a much higher slope than that of the SEAD encoder. Fig. 2.7 shows a wider range of values of K, and is scaled to highlight the complexity evolution of the SEAD encoder, see the orange and yellow lines labeled "SEAD-TOP Update/Merge" and "SEAD-TOP Patiton," and the partition operations of the SED encoder, see magenta curve labeled "SED Partition." These curves showcase the low

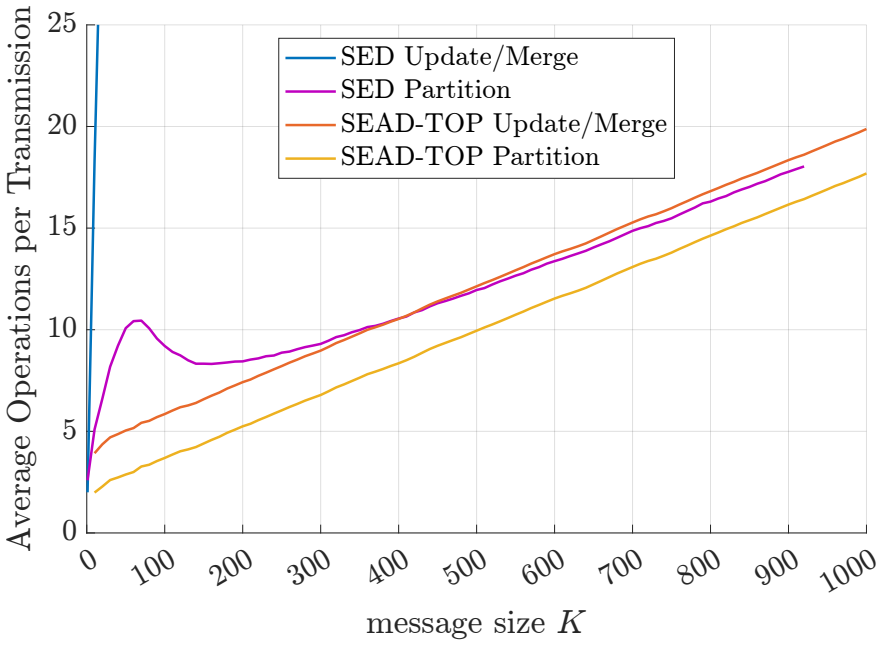

Figure 2.7: Average number of posterior groups operated on at each symbol transmission as a function of message size K, same as Fig. 2.6, scaled to highlight the curves for the SEAD encoder with thresholding of order posteriors, and for a wider range of K values. The solid blue curve with a very high slope, labeled "SED Update/Merge" shows the update and merge operations for the SED encoder; the solid magenta curve with a lobe, labeled "SED Partition," shows the partition operations for the SED encoder; the solid orange line labeled "SEAD-TOP Update/Merge," shows the update and merge operations for the SEAD encoder with TOP partitioning; and the solid yellow curve labeled "SEAD-TOP Partition" shows the partition operations for the SEAD encoder with TOP partitioning. These are curves are the same simulations in Fig. 2.6 for a channel with capacity $C = 0.50$.

complexity of the SEAD encoder with TOP partitioning. Note that when $K = 1000$, an average of only 20 posterior groups are visited at each transmission.

Quadratic Lines were fitted in Fig. 2.9, to each curve in Fig. 2.8, to compare the analytical result that the complexity of the SPM algorithm is of quadratic order, with the simulated complexity. The quadratic coefficient, about 0.0154, of the two cures for the SEAD encoder with TOP partitioning are two orders of magnitude below that of the SED encoder 1.5, which demonstrates the lower complexity of the SEAD encoder with TOP partitioning. For this encoder, the complexity that from the linear term is higher than that of the quadratic term, up to $K = 300$. These results show that complexity of the SPM-TOP algorithm allows

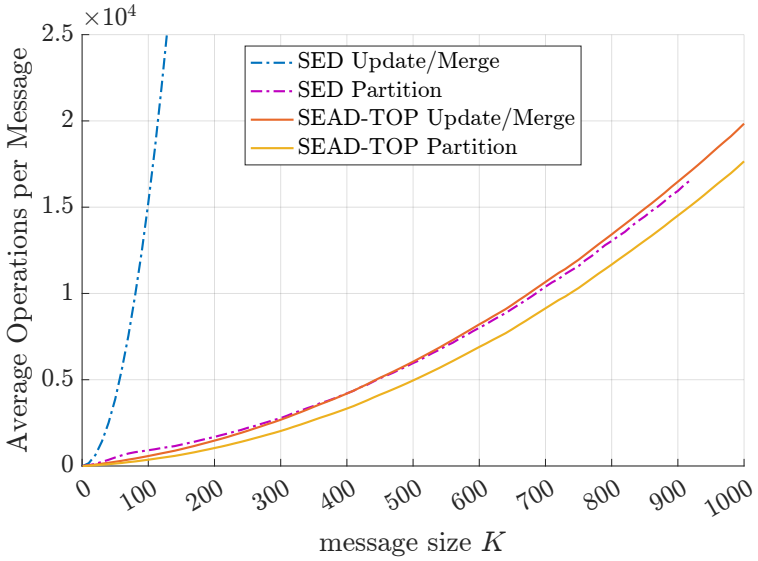

Figure 2.8: Average number of posterior groups operated on per message transmission as a function of message size K, over a channel with capacity $C = 0.50$. The solid blue curve labeled "SED Update/Merge" shows the update and merge operations for the SED encoder; the solid magenta curve labeled "SED Partition" shows the number of partition operations for the SED encoder; the solid orange curve labeled "SEAD-TOP Update/Merge" shows the update/merge operations for the SEAD encoder with TOP partitioning; and the solid yellow curve labeled "SEAD-TOP Partition" shows the partition operations for the SEAD encoder with TOP partitioning. These curves show the same simulation in Fig. 2.6 and Fig. 2.7, but instead of operations per symbol, the vertical axis is the number of operations per message.

for fast execution time. The quadratic curves fit very well each of the simulated curve, and thus validate the theory that the complexity order, as a function of blocklength, is linear for each transmission, and is quadratic for the entire block.

The simulations in Fig. 2.3 and Fig. 2.4 of the SED encoder consisted of 10^4 of trials for channels with capacities $C = 0.10$ and 0.25 and 10^5 trials for the other channels; the simulation of the SEAD encoder consisted of 10^5 trials for channels with $C = 0.1$ and 0.25 and 10^6 trials for the other channels. All the complexity simulations consisted of 10^5 trials for each value of $K = 1, 10, 20, \ldots, 1000$, and for a channel with capacity $C = 0.50$. The decoding threshold used in all simulations was $\epsilon = 10^{-3}$. The simulations were performed on a 2019 MacBook Pro laptop with a 2.4 GHz, 8-core $i9$ processor and 16 GB of RAM, and with transmitter and receiver operating alternatively on the same processor.

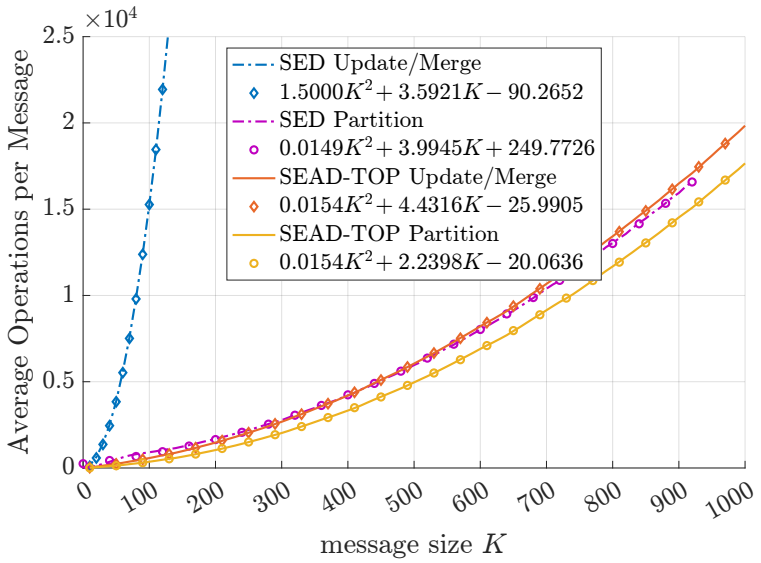

Figure 2.9: Shows parabolas fitted to each curve in Fig. 2.9. The solid lines are the same curves of Fig. 2.9, and the markers on each curve show a quadratic line fitted to the curve. The legend shows the markers and the quadratic curves $aK^2 + bK + c$ fitted to each curve. The figure highlights the much lower quadratic coefficient for the SEAD encoder with TOP partitioning, compared to the SED encoder.

2.7 Conclusion

To reduce complexity, the simulation framework in this dissertation replaces SED with the small enough *absolute* difference (SEAD) partitioning constraints, that relax the SED constraints by allowing the negative side, making it a two-sided version of SED during the communication phase. The SEAD constraints admit a partitioning rule that only applies a threshold to the vector ordered posteriors, the TOP rule. In this way, SEAD allows a low complexity approach that organizes messages according to their type, i.e. their Hamming distance from the received word, orders messages into groups according to their posterior, and partitions the messages with a simple threshold without requiring any swaps. This partitioning requires splitting at most one posterior per transmission, and greatly simplifies the process of updating and reordering the posterior groups into a sorted list, after each transmission.

The simplified SEAD-TOP encoder has a complexity order of $O(K^2)$, and allows simulations for message sizes of at least 1000 bits. The high rate performance attained with the SEAD encoding rule motivates the analysis shown in Chapter 3. From a practical perspective, the simulation results themselves provide new lower bounds on the achievable rates possible for the BSC with full feedback. For example, with an average block size of 200.97 bits corresponding to $K = 99$ message bits, simulation results for a target codeword error rate of 10^{-3} show a rate of $R = 0.493$ for the channel with capacity $C = 0.5$, i.e. 99% of the capacity.

CHAPTER 3

Analytical Bounds

This chapter analyzes achievability bounds for the system model considered in Chapter 2. Through a new analysis, this chapter proves that the SEAS partitioning rule in 2.2 guarantees an expected rate $K/\mathsf{E}[\tau]$ above the highest bound developed for the system model, the bound by Yang *et al.* [YPA21] that is achieved by Naghshvar's SED encoder. The new analysis facilitates further tightening of the bound by Yang *et al.* This chapter generalizes the concept of a "surrogate" process in [YPA21] and construct a matching surrogate process for the SEAD encoder.

3.1 Encoder and Decoder Model

This chapter studies analytical bounds for the model in chapter 2, and thus the system model is the same, see Fig. 3.1. At each time $t = 1, 2, \ldots, \tau$ the encoder sends a binary symbol $X_t \in \{0, 1\}$ to the decoder over the forward BSC channel, and the received symbol $Y_t \in \{0, 1\}$ is sent back to the encoder over the noiseless feedback channel before the next symbol is encoded. The communication problem consists of transmitting a K-bit information sequence Θ, from the encoder to the decoder, using the smallest expected number of symbols $\mathsf{E}[\tau]$, and with error probability $\Pr(\hat{\Theta} \neq \Theta)$ bounded by a small threshold ϵ. The information sequence, or message, Θ is sampled from the message space Ω according to some probability distribution $\mathcal{P}_\Omega$, that is $\Theta \sim \mathcal{P}_\Omega(\Omega)$. Throughout this dissertation, a Random Variable (R.V.) is denoted by an upper case letter, and realizations with lower case letters. For most of this dissertation, the sample space Ω is $\{0, 1\}^K$, where K is the message length,

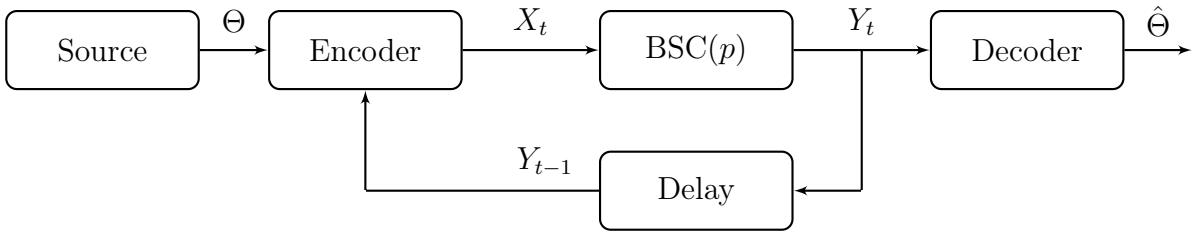

Figure 3.1: Model for sequential transmission over the BSC with noiseless feedback. The model consists of an information source, an encoder, a noisy forward channel, a decoder and a noiseless feedback channel. The model is the same used in chapter 2, Fig. 2.1. The encoder needs to communicate to the decoder a K-bit message Θ from a source. At each time $t = 1, 2, \ldots, \tau$, the encoder sends a binary symbols X_t over the forward channel, the BSC with a crossover probability p, and the decoder receives a binary symbol Y_t, that is a noisy version of X_t, according to the channel transition function $\Pr(Y_t \neq X_t) = p$. The channel symbol Y_t is subsequently sent back to the encoder via the noiseless feedback channel, and is available to the encoder to encode the next and future symbols $X_{t+1}, X_{t+2}, \ldots, X_\tau$. The transmission ends when any message i attains a posterior probability $\Pr(i = \Theta \mid Y^t) \geq 1 - \epsilon$, for a small threshold ϵ, and the message i becomes the estimate $\hat{\Theta}$ of Θ.

and the probability distribution is uniform, denoted by $\mathcal{U}$; that is, $\Theta \sim \mathcal{U}(\Omega)$. In this chapter the information sequence Ω is assumed to be available to the encoder, from the start of the transmission. The encoder computes each next transmitted symbol X_{t+1} using the message Θ, the feedback sequence $Y^t = Y_1, Y_2, \ldots, Y_t$, and possibly the previous symbols $X^t = X_1, X_2, \ldots, X_t$. The decoder and the encoder both compute the posterior probabilities $\rho_i(y^t)$, introduced in chapter 2, equation (2.3), defined by $\rho_i(y^t) = \Pr(\Theta = i \mid Y^t = y^t)$. The process terminates at the earliest time $t = \tau$ that a candidate $i \in \Omega$ achieves a posterior $\rho_i(y^\tau) \geq 1 - \epsilon$, and the message i that attains such a posterior becomes the decoder estimate $\hat{\Theta}$. For each $i \in \Omega$, let $U_i(t)$ denote the log likelihood ratio defined in Eq. (2.19). The stopping time τ introduced in (2.1) is equivalent to:

$$\tau \triangleq \min_{t \in \mathbb{N}} \{ \exists i \in \Omega : U_i(t) \geq \log_2 \left(\tfrac{1-\epsilon}{\epsilon} \right) \}. \tag{3.1}$$

The expected rate is defined by:

$$\text{Rate} \triangleq \frac{K}{\mathsf{E}[\tau]}\,. \tag{3.2}$$

When the number of K-bit messages transmitted is n, the total information length is $n \cdot K$-bits, the total number of transmitted symbols is $n\bar{\tau}$-bits and the overall rate is $\frac{nK}{n\bar{\tau}} = \frac{K}{\bar{\tau}}$. In the limit the average $\bar{\tau}$ becomes $\mathsf{E}[\tau]$, and the expected rate $\mathsf{E}[R]$ is of the form given in (3.2). The analytical achievability bounds in this dissertation are computed using alternative stopping rules that upper bound the stopping time τ in (3.1). The lower bound on τ and upper bound on the rate use a stopping time that lower bounds the definition of τ in (3.1).

3.2 Achievability Bound Problem Statement

The communication problem consists of finding the highest lower bound on the expected achievable rate R, given by $\mathsf{E}[R] \triangleq K/\mathsf{E}[\tau]$, with bounded error rate: $\mathsf{E}\big[\mathbb{1}_{\hat{\Theta}\neq\Theta}\big] = \Pr(\hat{\Theta} \neq \Theta) \leq \epsilon$, as well as a scheme that achieves the rate lower bound. The sequential problem can be described by:

$$\begin{aligned} &\text{minimize} && \tau_B && (3.3)\\ &\text{subject to} && \mathsf{E}[\tau] \leq \tau_B && (3.4)\\ & && \Pr(\hat{\Theta} \neq \Theta) \leq \epsilon && (3.5) \end{aligned}$$

3.3 Previous Results

To prove that the SED scheme of Naghshvar *et. al.* [NJW15] is a posterior matching BSC scheme as described in [SF11], it suffices to show that the scheme uses the same encoding function as [SF11] applied to a permutation of the messages. Since the posteriors $\rho_i(y^t)$ are fully determined by the history of received symbols Y^t, a permutation of the messages can

be defined concatenating the messages in S_0 and S_1, each sorted by decreasing posterior. This permutation induces a c.d.f. on the corresponding posteriors. Then, to satisfy the *posterior matching principle*, the random variable U could just be the c.d.f. evaluated at the last message before θ. The resulting encoding function is given by $X_{t+1} = 0$ if $U < 1/2$, otherwise $X_{t+1} = 1$.

3.4 Achievable Rate by Yang *et al.*

Naghshvar *et al.* [NJW15] proposed the "posterior matching" communication scheme analyzed in this section, with the SED partitioning rule (2.9), given by:

$$\text{SED rule: } 0 \leq \sum_{i \in S_0} \rho_i(y^t) - \sum_{i \in S_1} \rho_i(y^t) < \min_{i \in S_0} \rho_i(y^t)\,. \tag{3.6}$$

This is the rule described as the "small enough difference" (SED) rule by Yang *et al.* [YPA21]. Naghshvar *et al.* proved that the communication scheme, with SED partitioning achieves the channel capacity asymptotically. An alternative prove that the scheme, under SED partitioning, achieves the channel capacity is to just show that it is in fact a "posterior matching" scheme. It suffices to show that the scheme uses the same encoding function as [SF11] applied to a permutation of the messages. Since the posteriors $\rho_i(y^t)$ are fully determined by the history of received symbols Y^t, a permutation of the messages can be defined by the ordering of the message that results from concatenating the messages in S_0 and S_1, each sorted by decreasing posterior. This permutation induces a c.d.f. on the corresponding posteriors. Then, to satisfy the "posterior matching principle", the random variable U could just be the c.d.f. evaluated at the last message before θ. The resulting encoding function is given by $X_{t+1} = 0$ if $U < 1/2$, otherwise $X_{t+1} = 1$.

Yang *et al.* [YPA21] developed a finite-blocklength lower bound on the rate achievable by the SED encoder. This bound is computed from an upper bound on the expected blocklength

$\mathsf{E}[\tau]$ for each message length K provided in [YPA21], equation (3.17). Yang's achievability bound is the highest existing lower bound that could be found for the model before the analysis in this dissertation was developed. The achievability bounds in this dissertation are also published in [AGW23].

The analysis by Yang *et al.* consists of splitting the single phase process by Naghshvar *et al.* [NJW15] into a two phase process, with a communication phase (or phase I) and a confirmation phase (or phase II). The communication phase starts at time $t = 0$ and ends at a stopping time T, when the transmitted message attains a log likelihood ratio $U_i(t)$ at or above 0 for the first time, defined in [YPA21] by:

$$\text{Phase I stopping time: } T \triangleq \min_{t\in\mathbb{N}}\{U_i(t) \geq 0 : \theta = i\}\,. \tag{3.7}$$

Note that $U_i(t) = 0 \iff \rho_i(y^t) = \frac{1}{2}$, and that $U_i(t) \in [0, C_2) \iff \rho_i(y^t) \in [\frac{1}{2}, q)$. The confirmation phase starts at time T, when the communication phase ends, and ends at the stopping time τ where $\rho_i(y^t) \geq \frac{1}{2}$, also where i is the transmitted message θ. This is a method first used by Burnahsev in [Bur76]. The two phase analysis allows one to exploit the tighter constraints that the process satisfies during the confirmation phase.

Let $\mathcal{F}_t \triangleq \sigma(Y^t)$, the σ-algebra generated by the sequence of received symbols up to time t, where $Y^t = [Y_1, Y_2, \ldots, Y_t]$. Yang *et al.* showed that the SED encoder from Naghshvar *et al.* [NJW15] guarantees that the following constraints (3.8)-(3.11) are met:

$$\mathsf{E}[U_i(t+1)|\mathcal{F}_t, \theta = i] \geq U_i(t) + C, \qquad \text{if } U_i(t) < 0, \tag{3.8}$$

$$|U_i(t+1) - U_i(t)| \leq C_2, \tag{3.9}$$

$$\mathsf{E}[U_i(t+1)|\mathcal{F}_t, \theta = i] = U_i(t) + C_1, \qquad \text{if } U_i(t) \geq 0, \tag{3.10}$$

$$|U_i(t+1) - U_i(t)| = C_2, \qquad \text{if } U_i(t) \geq 0\,. \tag{3.11}$$

Meanwhile, Naghshvar *et al.* showed that the SED encoder also satisfies the stricter con-

straint that the average log likelihood ratio $\mathbf{U}(t)$, as defined in equation (3.12), and is also a submartingale that satisfies equation (3.13), which is equivalent to (3.14):

$$\mathbf{U}(Y^t) \triangleq \sum_{i=1}^{M} \rho_i(Y^t) U_i(Y^t) \tag{3.12}$$

$$\mathsf{E}[\mathbf{U}(Y^{t+1}) \mid \mathcal{F}_t] \geq \mathbf{U}(Y^t) + C \tag{3.13}$$

$$\mathsf{E}\left[\sum_{i=1}^{M} \left(\rho_i(y^{t+1}) U_i(t+1) - \rho_i(y^t) U_i(t) \right) \middle| \mathcal{F}_t \right] \geq C\,. \tag{3.14}$$

The process $\mathbf{U}(t)$ is a weighted average of values $U_i(t)$, some of which increase and some of which decrease after the next transmission $t+1$.

Yang *et al.* [YPA21] analyzed the expectation $\mathsf{E}[T]$ as a martingale stopping time, on the process $\zeta(t)$ defined by $\zeta(t) \triangleq \frac{U_i(t)}{C} - t$. The bound by Yang *et al.* required that inequalities (3.8) and (3.9) be satisfied, so that the process $\zeta(t)$ would exhibit the desired properties.

Before continuing with the analysis by Yang *et al.* [YPA21], the use of the restriction to event $\{\theta = i\}$, in the communication phase stopping time, is explained. The analysis only terminates the process in the confirmation phase, which starts at time T, when the correct message attains $U_i(t) \geq 0$. For this reason, the restriction is also "inherited" by the process stopping time τ. When applied to the communication phase stopping time T, the restriction to the event $\{\theta = i\}$ facilitates the analysis. When applied to the overall stopping time τ, it defines a "Gene" aided decoder that prevents early decoding when the wrong message $i \neq \theta$ attains a posterior $\rho_i(y^t) \geq 1 - \epsilon$. In reality, the process ends when any message $i \in \Omega$ attains $\rho_i(y^t) \geq 1 - \epsilon$. The "Gene" aided decoder may force the process to continue past the original stopping time τ, and only stop when the posterior of message θ attains the stopping condition. Thus, the "Gene" aided decoder exhibits a stopping time τ that upper bounds that of the original decoder that is analyzed.

Yang *et al.* [YPA21], [YW19] upper bounded the confirmation phase expected time $\mathsf{E}[\tau - T]$ using a Markov Chain analysis that exploits the larger and fixed magnitude step size of

C_2 during the confirmation phase, see (3.11), and the fixed expected step size in equation (3.10). It is possible that the correct message enters the confirmation phase, and then falls back to the communication phase. This "fall back" event is formally described by:

$$\text{Fall back event: } \{t_2 > t_1 = T\} \cap \{U_i(t_1) \geq 0\} \cap \{U_i(t_2) < 0\} \cap \{\theta = i\} \tag{3.15}$$

Since T is the time of the first crossing into the confirmation phase, the time $\tau - T$, which is analyzed by Yang *et al.* using a Markov Chain, includes the time that the message θ takes to return to the confirmation phase if a "fall back" occurs.

Yang *et al.* used the two phase analysis to obtained the following upper bounds τ_B on the expected blocklength $\mathsf{E}[\tau]$:

$$\tau_B = \frac{\log_2(M-1) + C_2}{C} + \left\lceil \frac{\log_2(\frac{1-\epsilon}{\epsilon})}{C_2} \right\rceil \frac{C_2}{C_1} + 2^{-C_2} \left(\frac{2C_2}{C} - \frac{C_2}{C_1} \right) \frac{1 - \frac{\epsilon}{1-\epsilon} 2^{-C_2}}{1 - 2^{-C_2}} \,. \tag{3.16}$$

Yang *et al.* analyzed a tighter upper bound on $\mathsf{E}[\tau]$, which is found in Thm. 7, Lemma 4 in [YPA21]. The tighter bound was obtained by synthesizing a surrogate martingale $U_i'(t)$ with stopping time T' that upper bounds T, which is a degraded version of the sub-martingale $U_i(t)$. The martingale $U_i'(t)$ guarantees that whenever $U_i'(t) < 0$, then $U_i'(t+1) \leq \frac{1}{q} \log_2(2q)$, while still satisfying the constraints needed to guarantee the bound (3.16). An achievability bound on the expected blocklength for the surrogate process, $U_i'(t)$, is constructed from (3.16) by replacing some of the C_2 values by $\frac{1}{q} \log_2(2q)$. The new bound from [YPA21] Lemma 4 is given by:

$$\tau_B \leq \frac{\log_2(M-1)}{C} + \frac{\log_2(2q)}{q \cdot C} + \left\lceil \frac{\log_2(\frac{1-\epsilon}{\epsilon})}{C_2} \right\rceil \frac{C_2}{C_1} + 2^{-C_2} \left(\frac{C_2 + \frac{\log_2(2q)}{q}}{C} - \frac{C_2}{C_1} \right) \frac{1 - \frac{\epsilon}{1-\epsilon} 2^{-C_2}}{1 - 2^{-C_2}} \,. \tag{3.17}$$

This bound also applies to the original process $U_i(t)$, since the blocklength of the process $U_i'(t)$ upper bounds that of $U_i(t)$. The bound (3.17) is lower because $\frac{1}{q} \log_2(2q)$ is smaller than C_2. The improvement is more significant as $p \to 0$ because $\frac{1}{q} \log_2(2q)$ grows from 0 to 1

as $p \to 0$, while C_2, instead, grows from 0 to infinity. The rate lower bound is given by $\frac{K}{\mathsf{E}[\tau]}$, where $\mathsf{E}[\tau]$ is upper bounded by (3.17) from Thm. 7 [YPA21]. We explain this analysis in detail in Thm. 2 and generalize it to other processes that are not restricted to martingales.

3.5 Motivation, Observations and Approach

This research seeks informative analytical bounds on the rates that are possible when communicating over the binary symmetric channel with full, noiseless feedback. To be informative, any lower bound on the rate must be higher than the lower bounds developed for less capable systems with limited feedback, such as the bound for stop feedback by Polyanskyi *et al.* [PPV11]. Note that the BSC with full, noiseless feedback is more capable than the BSC with stop feedback, which is itself more capable than the BSC without feedback. Consider "downgrading" the BSC with full feedback to the stop feedback system by adding a "black box" containing a replica of the receiver, that outputs a true-false symbol informing the transmitter when the process stops. Then, the achievable performance of the BSC with full and noiseless feedback is lower bounded by that of the BSC with stop feedback. This dissertation seeks to analyze a finite-blocklength achievability bound, for the BSC with full and noiseless feedback, that is higher than the highest achievability bound for the BSC with stop feedback, by Polyanskyi *et al.* [PPV11].

The new analysis presented in this chapter was originally motivated by simulations that achieved excellent performance with relaxed encoding rules that violated the constraints of previous analysis.

- Performance results from simulations that enforced either relaxation of the SED rule, with less restrictive partitioning, demonstrated performance that was indistinguishable from that of the original strict SED rule. The SEAD rule has the significant advantage of lower complexity implementations, e.g. it admits the simple partitioning by thresholding of ordered posteriors, see 2.3, which then allows updating the posteriors

and merging S_0 and S_1 into a sorted list with much fewer operations. Specifically, the relaxed rules can accomplish partitioning while splitting at most one posterior group at a given time t.

- A notable property of the SEAD partitioning rule is that it guarantees the following inequality: $\mathsf{E}[U_\theta(t+1) - U_\theta(t) \mid Y^t = y^t] \geq C$. This is an expectation that only requires knowledge of the posteriors obtained from the received sequence Y^t, which is available to both encoder and decoder at any time t. The expectation $\mathsf{E}[U_\theta(t+1) - U_\theta(t) \mid Y^t = y^t]$ is a function of all the posteriors in the space Ω. The SED rule satisfies the more strict constraint $\forall\, i \in \Omega :\ \mathsf{E}[U_i(t+1) - U_i(t) \mid Y^t = y^t, \theta = i] \geq C$, (3.8).

- The problem statement seeks a bound on $\mathsf{E}[\tau]$, where $\mathsf{E}[\tau] = \sum_{i \in \Omega} \mathsf{E}[\tau \mid \theta = i] \Pr(\theta = i)$. The SED rule guarantees bound (3.17) on $\mathsf{E}[\tau \mid \theta = i]$ for every $i \in \Omega$, using constraint (3.8), and thus on $\mathsf{E}[\tau]$. However, the bound derived from the SEAD rule is not for every particular $i \in \Omega$, but is rather an expectation over the entire space Ω. This dissertation explores using the lower bound taken over the entire message space, a bound on $\mathsf{E}[U_\theta(t+1) - U_\theta(t) \mid Y^t = y^t]$, to analyze a bound directly on $\mathsf{E}[\tau]$. This bound may not apply to the expected decoding time condition on an individual message (that is, $\mathsf{E}[\tau \mid \theta = i]$ for a particular $i \in \Omega$) but rather holds for $\mathsf{E}\left[\mathsf{E}[\tau \mid \theta = i]\right] = \sum_{i \in \Omega} \mathsf{E}[\tau \mid \theta = i] \Pr(\theta = i)$.

3.6 Contributions

The contribution of this chapter are the following:

- This chapter shows achievability rate bounds [AGW23] for the BSC with noiseless feedback above previously developed bounds like those in [YPA21]. The achievabilty bounds meet the desired criteria of exceeding, even for very high rates, the bounds developed for the BSC with stop feedback that prevent the encoder from using the

received sequence to optimize the transmissions. The new bounds show the provable performance gain by the full, noiseless feedback over stop feedback.

- The achievability rate bounds apply to both the SED encoder analyzed in [YPA21], and to the small-enough-*absolute*-difference (SEAD) encoder in [AGW23].

- This chapter proposes a tighter communication phase stopping time analysis via a "surrogate process," also shown in [YPA21], and generalized in [AGW23]. The "surrogate process" analysis leads to a tighter stopping time bound for a class of processes with increasing expected step size that is not fully utilized in the bound of the stopping time of the original process.

- This chapter provides a new analysis framework for posterior matching over the BSC that forgoes submartingale analysis in the communication phase. This allows it to demonstrate that the achievable rate of [YPA21] can be attained with a broader set of encoders that satisfy less restrictive criteria than the SED constraint, like the SEAD 2.2 or the Weighted Median partitioning rule of 5.6.

- This chapter proves that the SEAD encoding rule from Chapter 2.2, which admits low complexity encoders like the systematic posterior matching algorithm 2.4, suffices to guarantee the same rate performance that has been previously established for SED encoders, e.g., [YPA21].

- This chapter proves that the systematic transmissions used in [AYW20], and in the SPM algorithm of chapter 2, to initially send the message symbols, meets the SED encoding rule, as well as the SEAD and "Weighted Median" encoding rule of chapter 5. During the systematic transmissions, operations are limited to simply storing the received sequence, further reducing the transmission complexity.

- This chapter provides a finite blocklength rate upper bound for the BSC with noiseless feedback under an error probability constraint $\Pr(\hat{\Theta} \neq \Theta) \leq \epsilon$. The rate upper bound

is below the channel capacity for a very large region, and is always below the epsilon capacity shown in [PPV11].

3.7 Organization

The rest of the chapter is organized as follows: Sec. 3.8 provide achievability rate bounds for the BSC with noiseless feedback with error rate constraints. Sec. 3.9 introduces a converse bound on the rate achievable by the BSC with noiseless feedback with a stopping rule of the form described by (3.1) that terminates the process only when a "reliable enough" estimate is obtained. Sec. 3.10 introduces eight Lemmas that are used to prove the theorems. Sec. 3.11 proves Thm. 1. Sec. 3.14 provides the proofs of Lemmas 1-6 . Sec. 3.12 proves Thm. 2 and Thm. 3. Sec. 3.13 proves the converse Thm. 4. Sec. 3.15 extends the achievability bounds in Thms. 1, 2, and 3 to arbitrary input distribution, and to the special case where systematic transmissions transform a uniform input distribution on $\Omega = \{0,1\}^K$ into a binomial distribution, given the received sequence Y^K. Sec. 3.16 provides plots of the bounds defined in Thms. 1, 2, 3, and 4 for forward BSC channels with capacities $0.90, 0.75, 0.50$ and 0.25, as a function of blocklength, along with simulation results that validate the expressions in the Thms. Sec. 2.7 provides the chapter's conclusion. Finally, Sec. 3.18 through Sec. 3.22 provides the proofs of some of the details and claims used, but not proven in the previous sections.

3.8 Achievability Theorems

The achievability bounds in this dissertation use the stopping rule τ introduced by Yang *et al.* [YPA21], and is given by:

$$\tau \triangleq \min\{t \in \mathrm{N} : \max_i\{U_i(t)\} \geq NC_2\}\,, \quad N \triangleq \left\lceil \frac{\log_2\left(\frac{1-\epsilon}{\epsilon}\right)}{C_2} \right\rceil\,. \tag{3.18}$$

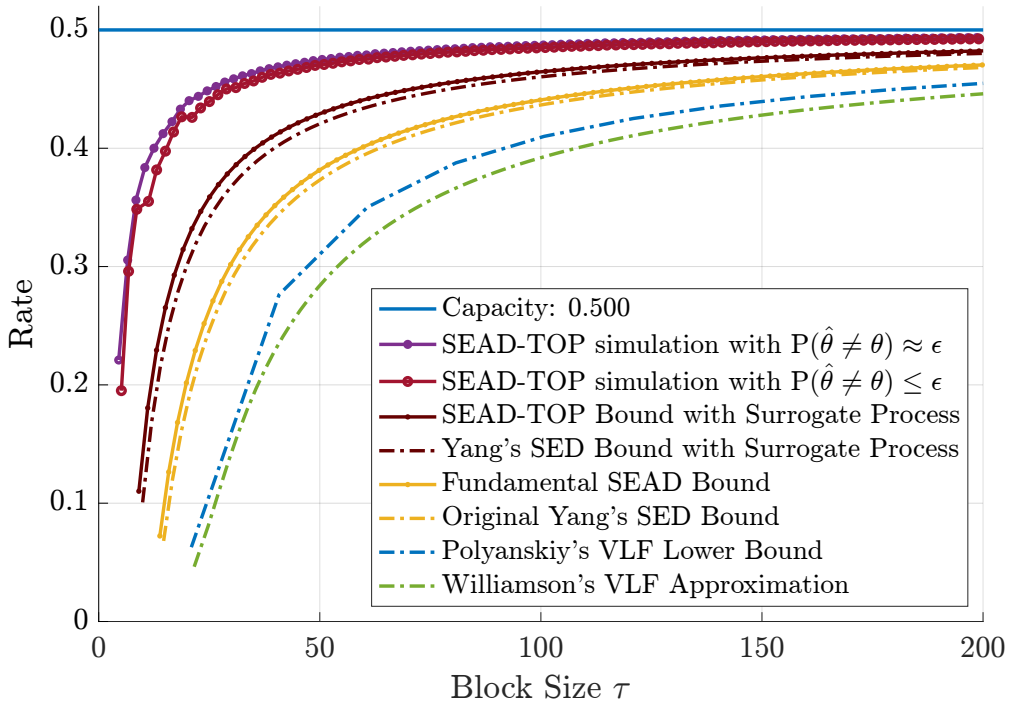

Figure 3.2: Achievability bounds vs. blocklength for a channel with capacity $C = 0.50$, shown by the horizontal, solid, blue line. The purple and red solid lines $-\bullet$ with circles show the simulated performance of the SEAD via the SPM-TOP algorithm of chapter 2. The red solid line is the bound derived with the new analysis and tightened using the surrogate process in Thms. 2 and 3. The dash dot line $-.$ is Yang's achievability bound from (3.17). The yellow solid line is the sub-optimal achievability bound of Thm. 1, and the yellow line $-.$ is Yang's original bound (3.16), not optimized with the surrogate process analysis. The blue dash dot line $-.$ shows Polyanskiy's [PPV11] achievability bound for variable-length, stop-feedback codes, and the green dash dot line $-.$ shows Williamson's approximation [Wil14] to Polyanskiy's [PPV11] achievability bound.

The new stopping rule replaces the original stopping time τ defined in (3.1), and facilitates the analysis of the confirmation phase. The upper bound on $\mathsf{E}[\tau]$, derived with the new stopping rule, also applies to the original stopping time. To show this, note that the new stopping time upper bounds the original stopping time. This is because $NC_2 \geq \log_2(1-\epsilon) - \log_2(\epsilon)$, thus, the new process does not stop after reaching a point $U_i(t) \in [\log_2(\frac{1-\epsilon}{\epsilon}), NC_2)$, while the original point does. However, if the new process stops, after reaching $U_i(t) \geq NC_2$, the original process also stops, because it has exceeded the $\log_2(\frac{1-\epsilon}{\epsilon})$ threshold.

The definition of the communication phase time T used in the achievability bounds is not

the stopping time used in by Yang *et al.* [YPA21]. Instead, it is the sum of stopping times used in [AGW23]. For each $n = 1, 2, 3, \ldots$, let T_n be the time at which the confirmation phase for message i starts for the n-th time (or the process terminates), and let $t_0^{(n)}$ be the time the encoder exits the confirmation phase for message i for the $(n-1)$-th time (or the process terminates). That is, for each $n = 1, 2, 3, \ldots$, let $t_0^{(n)}$ and T_n be defined recursively by $t_0^{(1)} = 0$ and:

$$T_n \quad = \min\{t \geq t_0^{(n)} : U_i(t) \geq 0 \,\text{or}\, t = \tau\} \tag{3.19}$$

$$t_0^{(n+1)} = \min\{t \geq T_n \;\; : U_i(t) < 0 \,\text{or}\, t = \tau\}\,. \tag{3.20}$$

Then, the total time the process $U_i(t)$ is not in its confirmation phase is given by:

$$T \triangleq \sum_{n=1}^{\infty} \left(T_n - t_0^{(n)}\right)\,. \tag{3.21}$$

The new definition of T is needed in the new analysis. Furthermore, it leads to a tighter analysis of the expected communication phase time. The definition of T_0 used in this dissertation, and in [AGW23], agrees with the communication phase time T in Yang *et al.* [YPA21].

The analysis of the bounds on $\mathsf{E}[\tau]$ follow the same approach used by Yang *et al.* where the time τ is split into $\tau = T + (\tau - T)$, where T is the new definition in (3.21). By linearity of expectations, $\mathsf{E}[\tau] = \mathsf{E}[T] + \mathsf{E}[\tau - T]$, which allows us to bound the terms $\mathsf{E}[T]$ and $\mathsf{E}[\tau - T]$ separately. Then, the bound τ_B on $\mathsf{E}[\tau]$ is obtained as the sum of the two bounds.

3.8.1 Fundamental Achievability Theorem

The next Thm. is the fundamental achievability Thm. that forgoes the submartingale analysis of the communication phase used by Yang *et al.* [YPA21].

Theorem 1. *Let $\epsilon < 1/2$ be a chosen bound on the frame error rate and let τ be a stopping time of a sequential transmission system over the BSC, defined in* (2.1)*. At each time t let*

the posteriors $\rho_1(Y^t), \rho_2(Y^t), \ldots, \rho_M(Y^t)$ *be as defined in* (7.1) *and the log likelihood ratios* $U_1(t), \ldots, U_M(t)$ *be as defined in* (7.2). *Suppose that for all times* t *for all received symbols* y^t, *and for each* $i \in \Omega$, *the constraints* (3.22)-(3.25) *are satisfied:*

$$\mathsf{E}[U_i(t+1) - U_i(t) | Y^t = y^t, \theta = i] \geq a\,, \quad \textit{where } a > 0\,, \tag{3.22}$$

$$U_i(t+1) - U_i(t) \leq C_2\,, \quad \textit{if } U_i(t) \leq 0\,, \tag{3.23}$$

$$\mathsf{E}[U_i(t+1) - U_i(t) | Y^t = y^t, \theta = i] = C_1\,, \quad \textit{if } U_i(t) \geq 0\,, \tag{3.24}$$

$$\mid U_i(t+1) - U_i(t) \mid = C_2\,, \quad \textit{if } U_i(t) \geq 0\,. \tag{3.25}$$

Suppose further that for all t *and* y^t *the following condition is satisfied:*

$$E[U_\theta(t+1) - U_\theta(t) | Y^t = y^t] \geq C \tag{3.26}$$

Then, expected stopping time $\mathsf{E}[\tau]$ *is upper bounded by* (3.27).

$$\mathsf{E}[\tau] \leq \frac{\log_2(M-1) + C_2}{C} + \left\lceil \frac{\log_2(\frac{1-\epsilon}{\epsilon})}{C_2} \right\rceil \frac{C_2}{C_1} + 2^{-C_2} \left(\frac{C_2}{C} - \frac{C_2}{C_1} \right) \frac{1 - \frac{\epsilon}{1-\epsilon} 2^{-C_2}}{1 - 2^{-C_2}}\,. \tag{3.27}$$

The sequential transmission process begins by randomly sampling a message θ from Ω. Using that selected message, at each time t until the decoding process terminates, the process computes an $X_t = x_t$, which induces a $Y_t = y_t$ at the receiver. The original constraint (3.8) used by Yang *et al.* [YPA21] requires that $\{U_i(t) - tC, \theta = i\}$ be a submartingale and allows for a bound on $U_i(t)$ at any future time $t + s$ for any possible selected message i, i.e. $\mathsf{E}[U_i(t+s) \mid \mathcal{F}_t, \theta = i] \geq U_i(t) + sC$. This is no longer the case with the new constraints in Thm. 1. While equation (3.22) of the new constraints make the process $U_i(t)$ a submartingale, it only guarantees that $\mathsf{E}[U_i(t+s) \mid \mathcal{F}_t, \theta = i] \geq U_i(t) + sa$ and a could be any small positive constant. The left side of equation (3.26) is a sum that includes all M realizations of the message; it is a constraint for each fixed time t and each fixed event $Y^t = y^t$ that governs the behavior across the entire message space Ω, and does not define a submartingale. For

this reason, the martingale analysis used by Naghshvar *et al.* in [NJW15] and Yang *et al.* in [YPA21] no longer applies.

A new analysis is needed to derive (3.27), the bound on the expected stopping time τ, using only the constraints of Thm. 1. This new analysis needs to exploit the property that the expected stopping time is over all messages, that is: $\mathsf{E}[\tau] = \sum_{i=1}^{M} \Pr(\theta = i)\mathsf{E}[\tau \mid \theta = i]$, which the original analysis does not need, because it guarantees that the bound (3.17) holds for each message individually, i.e., the bound holds on $\mathsf{E}[\tau \mid \theta = i]\,,\ i = 1, \ldots, M$. Note, however, that the original constraint (3.8) does imply that the new constraints are satisfied, so that the results derived in this chapter also apply to the setting of Naghshvar *et al.* [NJW15], and Yang *et al.* [YPA21]. The new constraints allow for a much simpler encoder and decoder design. This simpler design motivates the new analysis that forgoes the simplicity afforded by modeling the process $\{U_i(t), \theta = i\}$ as a martingale. The new achievability analysis includes in the communication phase time T every time interval where the transmitted message is not in its confirmation phase, that is, the time intervals where $\{U_i(t) < 0, \theta = i\}$. Specifically, in the event where $U_i(t_1) \geq 0$ for some t_1 and $\exists t > t_1$ with $U_i(t) < 0$, the time t is counted in the communication phase time T. The analysis by Yang *et al.* [YPA21] included these events in the confirmation phase time $\tau - T$ instead.

3.8.2 A "Surrogate Process" that Tightens the Achievability Bounds

The method used to obtain the achievability bounds (3.16) and (3.27), introduces a large sub-optimal term $\frac{C_2}{C}$, when the expectation $\mathsf{E}[\tau]$ is split into $\mathsf{E}[T]$ and $\mathsf{E}[\tau - T]$ to bound them separately. This term makes the bounds loose compared to (3.17). Since the expectation $\mathsf{E}[U_i(T)]$, the value of $U_i(t)$ at the end of the communication phase, could not be computed directly, the bound C_2 on $U_i(T)$ from (3.23) is used in the bound on the communication phase time $\mathsf{E}[T]$. However, this large value C_2 is not strictly needed to satisfy any of the constraints in Thm. 1. To overcome this sub-optimality, this dissertation proposes analyzing a surrogate process, which is a degraded version of the original process $U_i(t)$. The surrogate

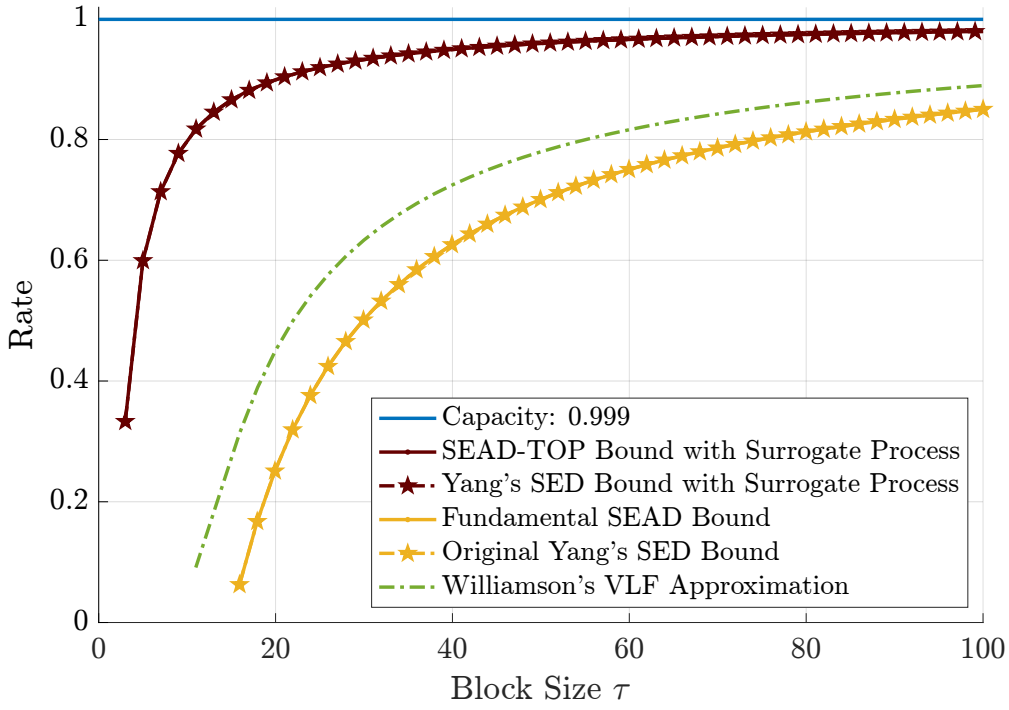

Figure 3.3: Achievability bounds vs. blocklength for a channel with capacity $C = 0.999$, shown by the horizontal, solid, blue line. The red solid line is the bound derived with the new analysis and tightened using the surrogate process in Thms. 2 and 3. The red dash dot line $-\star-\cdot$ with stars on top of the solid red line is Yang's achievability bound from (3.17), also using a surrogate process analysis. The yellow solid line is the sub-optimal achievability bound of Thm. 1, and the yellow line $-\star-\cdot$ with stars on top of the yellow solid line is Yang's original bound (3.16), not optimized with the surrogate process analysis. Note that the latter two fall below Williamson's approximation [Wil14] to Polyanskiy's [PPV11] achievability bound for for variable-length, stop-feedback codes.

process is a degradation in the sense that it is always below the value of the original process $U_i(t)$. For the surrogate process, the value at the start of the confirmation phase is bounded by a constant that is smaller than C_2.

The utility of the surrogate process may be better understood through the following frog-race analogy, illustrated in Fig. 3.4. A frog f_1 traverses a race track of length L jumping from one point to the next. The distance traveled by frog f_1 in a single jump is upper bounded by u_1. The jumps are not necessarily IID, but the expected progress of each jump is guaranteed to be bounded by l. However, not every jump is forward; f_1 may sometimes

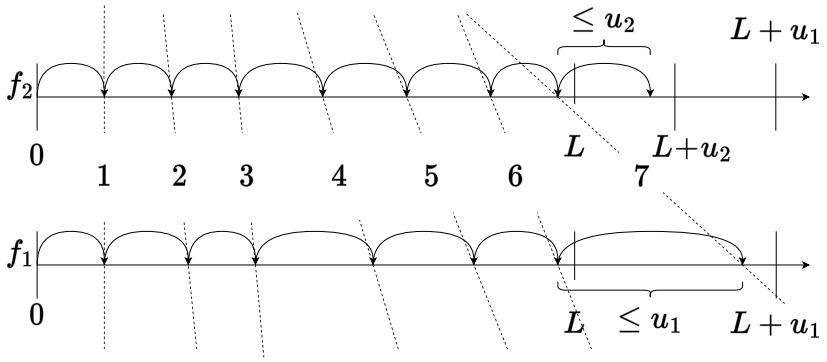

Figure 3.4: Example: frogs f_1 and f_2 jumping from 0 to L. The length of a single jump by f_1 is at most u_1. Frog f_2 jumps at the same times as f_1; however, the length of a single jump by f_2 is at most $u_2 < u_1$. This restriction forces frog f_2 to be always behind f_1, and thus reach L no sooner than frog f_1.

jump backwards. With only this information, an upper bound on the average number of steps that frog f_1 takes to reach the end of the track needs to be determined. This could be done using Doob's optional stopping theorem [Dur19b] to compute the upper bound as $\frac{L+u_1}{l}$, the maximum distance $L+u_1$ traveled from the origin to the last jump divided by the lower bound on average distance l of a single jump.

Perhaps this bound can be improved. The final point is located between L and $L+u_1$ and is reached in a single jump from a point between $L-u_1$ and L. If, for instance, the frog was restricted to only forward jumps, the term u_1/l could just be replaced by 1. This is not the case, as the process $U_i(t)$ does take steps backwards with probability p. Another property of $U_i(t)$ is exploited instead, which is that the maximum step size C_2 is not needed to guarantee the lower bound C on the average step size. Suppose now that a surrogate frog f_2 participates in the race along f_1, but with the following restrictions:

1. f_1 and f_2 start in the same place and always jump at the same time.

2. f_2 is never ahead of f_1, i.e. when f_1 jumps forward, f_2 jumps at most as far, and when f_1 jumps backwards, f_2 jumps at least as far.

3. Moreover, the forward distance traveled by frog f_2 in a single jump is upper bounded

by $u_2 < u_1$.

4. Despite its slower progress, the surrogate frog f_2 still satisfies the property that the expected length of each jump is lower bounded by l.

The average number of steps taken by f_2 will be upper bounded by $\frac{L+u_2}{l}$, also by Doob's optional stopping theorem. Since f_2 is never ahead of f_1, then f_2 crossing the finish line implies that f_1 has as well. Thus, $\frac{L+u_2}{l}$ is also an upper bound on the average number of jumps required for frog f_1 to reach across L.

The concept of the "surrogate process" is formalized in the next Thm, where the "surrogate" frog f_2 is replaced by the process $U_i'(t)$, the track length L is $\log(M-1)$, the maximum step size u_1 of the original process $U_i(t)$ is C_2, the maximum step size u_2 of the surrogate process $U_i'(t)$ is $B < C_2$, and the shared lower bound l on the expected step size is C.

Theorem 2: Surrogate Process Theorem. *Let the surrogate process $U_i'(t)$ be a degraded version of $U_i(t)$ that still satisfies the constraints of Thm. 1. Initialize the surrogate process as $U_i'(0) = U_i(0)$, and reset $U_i'(t)$ to $U_i(t)$ at every $t = t_0^{(n)}$, that is, at each t that the encoder exits a confirmation phase round for message i. Define $T_n' \triangleq \min\{t \geq t_0^{(n)} : U_i'(t) \geq 0 \text{ or } t = t_0^{(n+1)}\}$. Suppose that for some $B < C_2$, the process $U_i'(t)$ also satisfies the following constraints:*

$$U_i(t) < 0 \implies U_i'(t+1) - U_i'(t) \leq U_i(t+1) - U_i(t) \tag{3.28}$$

$$U_i'(t) < 0 \implies U_i'(t+1) \leq B \tag{3.29}$$

$$U_i'(T_n') - \frac{p}{q}\left(U_i(T_n) - C_2\right) \leq B. \tag{3.30}$$

Then, the total time $U_i'(t)$ is not in its confirmation phase is given by $T' \triangleq \sum_{n=1}^{\infty}\left(T_n' - t_0^{(n)}\right)$, and $E[T]$ is bounded by:

$$\mathsf{E}[T] \leq \mathsf{E}[T'] \leq \frac{B}{C}\left(1 + 2^{-C_2}\frac{1-2^{-NC_2}}{1-2^{-C_2}}\right) - \frac{\mathsf{E}[U_i(0)]}{C}. \tag{3.31}$$

Note that $T_n \leq T'_n$ for all n, because the definition of $U'_i(t)$ and constraint (3.28)*, demand that $U_i(t) \geq U'_i(t)$, thus $T \leq T'$. Also note that after the process terminates at the stopping time τ, both T_n and $t_0^{(n)}$ are equal to τ, which makes their difference* 0*. Then, the communication phase times T and T' are a sum of finitely many non-zero values.*

Theorem 3. *Consider sequential transmission over the BSC with noiseless feedback with an encoder that enforces the Small Enough Absolute Difference (SEAD) encoding constraints, used in Chapter 2, Sec. 2.2, and described by equations* (3.32) *and* (3.33) *below:*

$$\left| \sum_{i \in S_0} \rho_i(y^t) - \sum_{i \in S_1} \rho_i(y^t) \right| \leq \min_{i \in S_0} \rho_i(y^t) \tag{3.32}$$

$$\rho_i(y^t) \geq \frac{1}{2} \implies S_0 = \{i\} \text{ or } S_1 = \{i\}\,. \tag{3.33}$$

Then, the constraints (3.25)-(3.26) *in Thm. 1 are satisfied, and a process $U'_i(t)$, $i = 1, \ldots, M$ as described in Thm. 2 can be constructed with $B = \frac{1}{q}\log_2(2q)$. Then, the following upper upper bound on $\mathsf{E}[\tau]$ applies to both $U'_i(t)$, and to the original process:*

$$\mathsf{E}[\tau] \leq \frac{\log_2(M-1) + \frac{\log_2(2q)}{q}}{C} + \frac{C_2}{C_1}\left\lceil \frac{\log_2\left(\frac{1-\epsilon}{\epsilon}\right)}{C_2} \right\rceil + 2^{-C_2}\frac{1 - \frac{\epsilon}{1-\epsilon}2^{-C_2}}{1 - 2^{-C_2}}\left(\frac{\log_2(2q)}{qC} - \frac{C_2}{C_1}\right). \tag{3.34}$$

Note that meeting the SEAD constraints guarantees that both sets S_0 and S_1 are non empty. This is because, if either set is empty, the other one is the whole space Ω, and the difference in (3.32) *is* 1*, which is greater than any posterior in a space with more than one element with non-zero posterior.*

3.9 Converse Theorem

The next theorem proposes an upper bound on the rate achievable for the BSC with noiseless feedback, under the same constraints in the problem statement of Sec. 3.2. The original transmission problem consists of maximizing the expected rate, defined by $K/\mathsf{E}[\tau]$, or mini-

mizing the expected decoding time $\mathsf{E}[\tau]$. A lower bound on the achievable expected decoding time $\mathsf{E}[\tau]$ defines an upper bound on the achievable expected rate $K/\mathsf{E}[\tau]$.

3.9.1 Converse Bound Problem Statement

The converse bound problem consists on finding the highest lower bound τ_B on the expected decoding time $\mathsf{E}[\tau]$ that can be achieved with any scheme that bounds the error probability by $\Pr(\hat{\theta} \neq \theta) \leq \epsilon$. The converse problem can be formally described by:

$$\begin{aligned} &\text{maximize} && \tau_B && (3.35)\\ &\text{subject to} && \tau_B \leq \mathsf{E}[\tau] && (3.36)\\ & && \Pr(\hat{\theta} \neq \theta) \leq \epsilon && (3.37) \end{aligned}$$

Theorem 4. *Let $N = \lfloor \frac{\log_2(1-\epsilon)-\log_2(\epsilon)}{C_2} \rfloor$. That is, N is at most the minimum number of confirmation phase steps needed to decode. Then, the bound τ_B below lower bounds the $\mathsf{E}[\tau]$:*

$$\tau_B \triangleq (1-2\epsilon)\frac{K-1}{C} + 2\epsilon\frac{K-1}{\log_2(2q)} - (1-2\epsilon)\frac{\log_2(2q)}{C} \tag{3.38}$$

$$+ \frac{1-2^{-(N+1)C_2}}{2(1-2^{-(N+1)C_2}) - (1-2^{-NC_2})}\frac{C_2}{C_1}\left(1 - 2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right) \tag{3.39}$$

$$+ \frac{1-2^{-(N+1)C_2}}{2(1-2^{-(N+1)C_2}) - (1-2^{-NC_2})}\frac{C_2}{C_1}\left(N - 2^{-NC_2}\frac{1-2^{-NC_2}}{1-2^{-2NC_2}}2N\right) \tag{3.40}$$

This time is given by three expressions: the time where every message is in the communication phase, given by (3.38)*; the time that a message i that is not the correct message θ spends in the confirmation phase before the correct message ever reaches the confirmation phase, given by* (3.39)*; and the time the process takes to stop after the correct message θ enters the confirmation phase for the first time, given by* (3.40)*. Note that incorrect decoding must be accounted for in the second and third case, terms* (3.39) *and* (3.40)*.*

3.9.2 Approximated of the Rate and Blocklength of the SEAD Encoder

Let $N = \lceil \frac{\log_2(1-\epsilon)-\log_2(\epsilon)}{C_2} \rceil$, as defined in (3.18). Then, the expected decoding time of an encoder that enforces either the SEAD or SED encoding rules, may be approximated by $\mathsf{E}[\tau] \approx \tau_A$, where τ_A is given by:

$$\begin{aligned}\tau_A \triangleq & \frac{K-1}{C} - \frac{\log_2(2q)}{C} \\ & + \frac{1-2^{-(N+1)C_2}}{2(1-2^{-(N+1)C_2})-(1-2^{-NC_2})}\frac{C_2}{C_1}\left(1-2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right) \\ & + \frac{1-2^{-(N+1)C_2}}{2(1-2^{-(N+1)C_2})-(1-2^{-NC_2})}\frac{C_2}{C_1}\left(N-2^{-NC_2}\frac{1-2^{-NC_2}}{1-2^{-2NC_2}}2N\right)\end{aligned} \tag{3.41}$$

This approximation is similar to the upper bound of Thm. 4, in that assumes an expected step size $\mathsf{E}[\log_2(\rho_\theta(y^{t+1})) - \log_2(\rho_\theta(y^t))|Y^t = y^t] = C$, during the communication phase. This is an upper bounds on the expected step size of $\log_2(\rho_i(y^t)$, see the proof of Thm. 4. However, the expression enforces an error probability upper bounded by ϵ, as it only allows the confirmation phase to stop when the posterior of the item in confirmation reaches or exceeds the threshold ϵ. This is how the algorithm is implemented in 2. An approximation of the rate for the same encoders is give by $\frac{K}{\tau_A}$.

3.10 Lemmas

The following five Lemmas are the key elements to prove the theorems:

Lemma 1: Expected times. *Let T be the total time the transmitted message spends in the communication (including the time another message $i \neq \theta$ may spend in the confirmation phase), and let T_n and $t_0^{(n)}$ be as defined in* (3.19) *and* (3.20)*. Define $T^{(n)} \triangleq T_n - t_0^{(n)}$. For $r \leq s \leq t$, let $y_{r:s}^t$ be $[y_1, \ldots, y_s]$, the string of y_i values from y_r to y_s. Define the sets $\mathcal{Y}_{(\tau>t)}^{\epsilon}$,*

$\mathcal{Y}^\epsilon$, *and* $\mathcal{Y}_i^\epsilon$ *by:*

$$\mathcal{Y}^\epsilon_{(\tau>t)} \triangleq \{y^t \in \{0,1\}^t \mid \rho_j(y^t_{1:s}) < 1-\epsilon,\ \forall j \in \Omega, s \le t\} \tag{3.42}$$

$$\mathcal{Y}^\epsilon \triangleq \cup_{t=0}^{\infty} \mathcal{Y}^{t,\epsilon}_{(\tau>t)} \tag{3.43}$$

$$\mathcal{Y}_i^\epsilon \triangleq \{y^t \in \mathcal{Y}^\epsilon \mid \rho_i(y^t) < \tfrac{1}{2}\}\,, \tag{3.44}$$

then, $\mathsf{E}[\tau]$ *and* $\mathsf{E}[T]$ *are given by:*

$$\mathsf{E}[T] = \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}_i^\epsilon} \Pr(Y^t = y^t \mid \theta = i) = \sum_{i=1}^{M} \sum_{y^t \in \mathcal{Y}_i^\epsilon} \Pr(Y^t = y^t, \theta = i)\,. \tag{3.45}$$

$$\mathsf{E}[\tau] = \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}^\epsilon} \Pr(Y^t = y^t \mid \theta = i) = \sum_{y^t \in \mathcal{Y}^\epsilon} \sum_{i=1}^{M} \Pr(Y^t = y^t, \theta = i) \tag{3.46}$$

Lemma 2. *Suppose constraints* (3.22), (3.25), *and* (3.26) *of Thm. 1 are satisfied, and define* $V_i(y^t)$ *and* $\Psi_i(y^t)$*:*

$$V_i(y^t) \triangleq \mathsf{E}[U_i(t+1) - U_i(t) \,|\, Y^t = y^t, \theta = i] \tag{3.47}$$

$$\Psi_i(y^t) \triangleq \mathsf{E}[\log_2(\rho_i(y^{t+1})) - \log_2(\rho_i(y^t)) | Y^t = y^t, \theta = i]\,, \tag{3.48}$$

Define the set $\mathcal{A}_\epsilon$ *by:*

$$\mathcal{A}_\epsilon \triangleq \{y^t \in \mathcal{Y}_i^\epsilon : \rho_j(y^t) < \frac{1}{2}\ \forall j = 1, \ldots, M\}\,, \tag{3.49}$$

where $\mathcal{A}_\epsilon$ *does not depend on* i*. Then, the following inequalities holds:*

$$C \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}_i^\epsilon} \Pr(Y^t = y^t \mid \theta = i) \le \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}_i^\epsilon} V_i(y^t) \Pr(Y^t = y^t \mid \theta = i) \tag{3.50}$$

$$C \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{A}_\epsilon} \Pr(Y^t = y^t \mid \theta = i) \ge \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{A}_\epsilon} \Psi_i(y^t) \Pr(Y^t = y^t \mid \theta = i)\,. \tag{3.51}$$

Claim 2. *For the communication scheme described in Sec. 3.3, the following are equivalent:*

(i) $\mid U_j(t+1) - U_j(t) \mid = C_2$

(ii) $S_0 = \{j\}$ *or* $S_1 = \{j\}$

This claim implies that for constraint (3.25) *to hold, the set containing item* j*, with* $U_j(t) \geq 0$*, must be a singleton.*

Proof. See Sec. 3.18 □

Claim 3 (The Confirmation Phase is a Discrete Markov Chain)**.** *Let the partitioning of* Ω *at time* $t = s$ *be* $S_0 = \{j\}$*,* $S_1 = \Omega \setminus \{j\}$*, and suppose* $Y_{s+1} = 0$*. If the partitioning at time* $t = s+1$ *is also* $S_0 = \{j\}$*,* $S_1 = \Omega \setminus \{j\}$*, the same partitioning of time* s*, and* $Y_{t+1} = 1$*, then for all* $i = 1, \ldots, M$*,* $\rho_i(y^{t+1}) = \rho_i(y^{t-1})$*, that is* $\rho_i(y^s) = \rho_i(y^{s+2})$*. Proof: see appendix 3.21*

Lemma 3. *Suppose that constraints* (3.22)*,* (3.25)*, and* (3.26) *of Thm. 1 are satisfied, as in Lemma 2, then the following equalities hold:*

$$\sum_{y^t \in \mathcal{Y}_i^\epsilon} \mathsf{E}[U_i(t+1) - U_i(t) \,|\, Y^t = y^t, \theta = i] \Pr(Y^t = y^t \,|\, \theta = i) = \sum_{n=1}^{\infty} \mathsf{E}[U_i(T_n) - U_i(t_0^{(n)}) \,|\, \theta = i] \,. \tag{3.52}$$

$$\sum_{y^t \in \mathcal{Y}_i^\epsilon} \mathsf{E}[\log_2 \left(\frac{\rho_i(y^{t+1})}{\rho_i(y^t)} \right) | Y^t = y^t, \theta = i] \Pr(Y^t = y^t \,|\, \theta = i) = \mathsf{E}[\log_2 \left(\frac{\rho_i(y^{T_0})}{\rho_i(y^0)} \right) \mid \theta = i] \,. \tag{3.53}$$

The left side of 3.52 is the inner sum of (3.50) *in Lemma* (2)

Lemma 4. *Let* ϵ *be the decoding threshold, and let the decoding rule be* (3.18)*. Define the fall back probability as the probability that the correct message* θ *returns to the communication phase, computed at the start of a confirmation phase. Then, the fall back probability is a constant* p_f *independent of the message* $i = 1, \ldots, M$*, independent of the number of previous confirmation phase rounds* n*, and is given by:*

$$p_f = 2^{-C_2} \frac{1 - 2^{-NC_2}}{1 - 2^{-(N+1)C_2}} \,. \tag{3.54}$$

Let the decoding rule be similar to rule (3.18)*, in the sense that* N *is fixed regardless of the value of* $U_i(T) \in [0, C_2)$*. Particularly,* N *could be computed by:*

$$N \triangleq \left\lfloor \frac{\log_2(\frac{1-\epsilon}{\epsilon})}{C_2} \right\rfloor . \tag{3.55}$$

In this case, $(N+1)C_2 > \log_2(\frac{1-\epsilon}{\epsilon}) \geq NC_2$*, which lower bounds the decoding time. Define the recovery probability as the probability that the wrong message returns to the communication phase, given that it has just crossed into the confirmation phase. Then, the recovery probability is a constant* p_r *given by:*

$$p_r = \frac{1 - 2^{-NC_2}}{1 - 2^{-(N+1)C_2}} , \tag{3.56}$$

is independent of which incorrect message $i \in \Omega \setminus \theta$ *has reached the confirmation phase, provided it is not* θ*, and independent of the number of previous correct or wrong confirmation phase rounds* n*.*

Conversely, suppose that the wrong message $i \in \Omega \setminus \theta$ *has crossed into the confirmation phase, with* $U_i(t) \in [0, C_2)$ *for some time* t*. Then, the probability that the wrong confirmation phase ends in wrong decoding at some time* $t' > t$*, where message* i *attains* $U_i(t') \geq NC_2$*, instead of a recovery, is given by* $1 - p_r$*:*

$$1 - p_r = 2^{-NC_2} \frac{1 - 2^{-C_2}}{1 - 2^{-(N+1)C_2}} \tag{3.57}$$

Corollary of Lemma 4. *Suppose that, at some time* $t = t_2$*, the confirmation phase for the correct message is at a step* n*, that is* $\{U_i(t_2) \in [nC_2, (n+1)C_2)\} \cap \{\theta = i\}$*, and suppose further that the confirmation phase ends the first time* τ_2 *that* $U_i(t)$ *reaches state* $n + m$ *(*m *states ahead of* n*), or when* $U_i(t)$ *reaches state zero (*n *states backwards, that is:*

$$\tau_2 \triangleq \min\{t > t_2 : U_i(t) < C_2 \text{ or } U_i(t) \geq (n+m)C_2\} . \tag{3.58}$$

Let p_m be the probability that such confirmation phase ends with $U_i(\tau_2) \geq (n+m)C_2$, and let p_n^- be the probability that it ends with $U_i(\tau_2) < C_2$. Then, p_m and $p_n^- = 1 - p_m$ are given by:

$$p_m = \frac{1 - 2^{-nC_2}}{1 - 2^{-(n+m)C_2}} \tag{3.59}$$

$$p_{n^-} = \frac{1 - 2^{-mC_2}}{1 - 2^{-(n+m)C_2}} 2^{-nC_2} \tag{3.60}$$

Furthermore, if $n = m = N$ then:

$$p_N = \frac{1 - 2^{-NC_2}}{1 - 2^{-2NC_2}} \tag{3.61}$$

$$p_{N^-} = \frac{1 - 2^{-NC_2}}{1 - 2^{-2NC_2}} 2^{-NC_2} \tag{3.62}$$

Lemma 5: Confirmation Phase Time. *Suppose that the decoding rule is the one defined in* (3.18)*. Let p_f be the fall back probability defined in 4, and let $\mathsf{E}[S_n]$ be the expected duration of the n-th confirmation round. Then, $\mathsf{E}[S_n]$ is the same for any $n = 1, 2, \ldots$, and is a constant $\mathsf{E}[S]$, given by:*

$$\mathsf{E}[S] = \frac{C_2}{C_1} \left((1 - p_f)N - p_f \right) , \tag{3.63}$$

that does not depend on n, that is: $\mathsf{E}[S_n] = \mathsf{E}[S] \quad \forall n = 1, 2, \ldots$.

Let the decoder be a "Gene aided" decoder that does not stop when the wrong message reaches the decoding criteria. Then, the expected number of occurrences of the confirmation phase is given by $\sum_{j=0}^{\infty} p_f^i = \frac{1}{1-p_f}$ times, and the total confirmation phase time $\mathsf{E}[\tau - T]$ is given by:

$$\mathsf{E}[\tau - T] = \mathsf{E}[S] \frac{1}{1 - p_f} = \left(N - \frac{p_f}{1 - p_f} \right) \frac{C_2}{C_1} = \left(N - 2^{-C_2} \frac{1 - 2^{-NC_2}}{1 - 2^{-C_2}} \right) \frac{C_2}{C_1} \tag{3.64}$$

Equation (3.64) *(with a few modifications) was proven in Yang* et al. *[YPA21], using a Markov chain analysis.*

Suppose instead, that the wrong message has just entered confirmation, and the decoding rule is similar to rule (3.18)*, in that the process terminates when the message i in confirmation reaches $U_i(t) \geq NC_2$, for some $N \in \mathbb{N}$. Let the wrong confirmation phase end with a recover, when the wrong message in confirmation falls back to the communication phase, or in wrong decoding, when it reaches the decoding criteria. Let the recovery probability be p_r as defined in Lemma 4. Then, the expected duration $\mathsf{E}[S_n^{bad}]$, of the n-th wrong confirmation phase round, is a constant $\mathsf{E}[S^{bad}]$, given by:*

$$\mathsf{E}[S^{bad}] = \frac{C_2}{C_1}\left(1 - 2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right), \tag{3.65}$$

and does not depend on n. That is: $\mathsf{E}[S_n] = \mathsf{E}[S]\ \forall n = 1, 2, \ldots$.

Corollary of Lemma 5. *Let the the setting be the one described in Corollary 1, in which the process is in the correct confirmation phase at some time $t = t_2$ and $i = \theta$, with $nC_2 \leq U_i(t_2) < (n+1)C_2$. This confirmation phase ends the first time $t = \tau_2$ that either i reaches the origin: where $U_i(t) < C_2$, or it reaches state $n+m$: where $U_i(t) \geq (n+m)C_2$, according to rule 3.58. Then, such confirmation phase has an expected duration $\mathsf{E}[S_{n+m}]$, given by:*

$$\mathsf{E}[S_{n+m}] = \frac{C_2}{C_1}\left(\frac{1-2^{-nC_2}}{1-2^{-(n+m)C_2}}(m+n)-n\right) = \frac{C_2}{C_1}\left(m-2^{-nC_2}\frac{1-2^{-mC_2}}{1-2^{-(n+m)C_2}}(m+n)\right). \tag{3.66}$$

In the particular case where $m = n = N$, then $\mathsf{E}[S_{2N}]$ is given by:

$$\mathsf{E}[S_{2N}] = \frac{C_2}{C_1}\left(2N\frac{1-2^{-NC_2}}{1-2^{-2NC_2}} - N\right) = \frac{C_2}{C_1}\left(N - 2N2^{-NC_2}\frac{1-2^{-NC_2}}{1-2^{-2NC_2}}\right). \tag{3.67}$$

This Corollary could apply to the case where reaching the origin is associated with a wrong decoding, and reaching $n+m$ is associated with a correct decoding.

Lemma 6. *Let p_f be the fall back probability in Lemma 4, and suppose that $U_i(0) < 0\ ,\ \forall i =$*

$1, \ldots, M$. *Then the expectation* (3.52) *in Lemma 3 is upper bounded by:*

$$\sum_{i=1}^{M} \Pr(\theta = i) \sum_{n=1}^{\infty} \mathsf{E}[U_i(T_n) - U_i(t_0^{(n)}) \mid \theta = i] \leq \sum_{i=1}^{M} \Pr(\theta = i) \left(\frac{p_f}{1-p_f} C_2 + C_2 - U_i(0) \right) \tag{3.68}$$

$$\leq 2^{-C_2} \frac{1 - 2^{-NC_2}}{1 - 2^{-C_2}} C_2 + C_2 - \mathsf{E}[U_i(0)] \,. \tag{3.69}$$

Lemma 7: Markov Chain state probabilities on wrong confirmation phase. *Suppose that the wrong message $i \neq \theta$ has entered the confirmation phase at some time t. Then, $U_i(t) > 0$ and $U_i(t-1) < 0$. In this case $y^t \in \mathcal{Y}_i \setminus \mathcal{A}_\epsilon$, and either $y^{t-1} \in \mathcal{A}_\epsilon$ or $\exists j \neq i$ such that $U_j(t-1) \geq 0$, in which case also $y^{t-1} \in \mathcal{Y}_i \setminus \mathcal{A}_\epsilon$. The message i in confirmation (correct or wrong) is the entire set S_0, making it a singleton $S_0 = \{i\}$. By claim 3, the wrong confirmation phase is also a Markov Chain with steps size C_2. In the wrong confirmation phase, since $\theta \neq i$, the transmitter sends $1, 1, \ldots$ until a recovery occurs (or the process ends in error). Suppose that the wrong Markov Chain has n states. Let r_0 denote the time spent at the starting state 0, and let r_i $i = 1, 2, \ldots, n$ denote the time spent at state i. Then, in the limit, as $n \to \infty$, each r_i is given by:*

$$r_{k+1} = r_0 \frac{p^{k+1}}{q^{k+1}}, \ \forall k = 1, 2, \ldots \,, \tag{3.70}$$

and $r_0 = \frac{1}{q}$. Note that, message i spends one time unit at state 0, if recovery happens in the first transmission. This immediate recovery happens with probability q, and with probability p state 1 is reached instead. A future return necessarily implies additional time in state 0. As $n \to \infty$, the message i returns to state 0 a.s., leading to $r_0 = \sum_{j=0}^{\infty} p^j = \frac{1}{1-p}$.

Lemma 8: Expected Sum of Bad Confirmation Steps. *Let S be the sum of all the steps that message θ takes, during any and all bad confirmation phases that may happen*

before an initial correct confirmation phase. Then, S is given by:

$$S = \frac{1}{C} \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}_i^\epsilon \setminus \mathcal{A}_\epsilon} E[\log_2(\rho_i(y^{t+1})) - \log_2(\rho_i(y^t)) | \theta = i, Y^t = y^t] \,. \tag{3.71}$$

Denote by m the number of consecutive times a wrong message enters confirmation before an initial correct confirmation. Let the crossing posterior of such message be ρ_m, given by $\rho_m = \Pr(\theta = i \mid Y^t = y^t)$, where i is the m-th incorrect message to enter the confirmation phase by time t, and before message θ enters for the first time. Note that $1 - \rho_m$ is the probability that message i is the incorrect message, given $\{Y^t = y^t\}$. Then, the value of S is given by:

$$S = \sum_{n=1}^{\infty} V_q(\rho_n) \Pr(m \geq n) \,, \tag{3.72}$$

where,

$$V_q(\rho_n) = \log_2 \frac{q}{q(1-\rho_n) + p\rho_n} \,. \tag{3.73}$$

And the probability $\Pr(m \geq n)$ is given by:

$$\Pr(m \geq n) = \prod_{r=1}^{n} (1 - \rho_r) \,. \tag{3.74}$$

This Lemma implies that the value of $\log_2(\rho_j(y^t)$, for a message $j \in \Omega$ not in confirmation, progresses by exactly $V_q(\rho_n)$, from the time another message enters the confirmation phase, to the time such message falls back to the communication phase.

3.11 Proof of Thm. 1

Proof. The proof uses linearity of expectations, $\mathsf{E}[\tau] = \mathsf{E}[\tau - T] + \mathsf{E}[T]$, to separately analyzed bounds first on $\mathsf{E}[T]$, and then $\mathsf{E}[\tau - T]$.

First Lemmas 1 is used to express $\mathsf{E}[T]$ as a sum of probabilities of events $y^t \in \mathcal{Y}_i^\epsilon$, over each $i \in \Omega$, and the Lemma 2 is used to bound the resulting sum of probabilities as follows:

$$\mathsf{E}[T] = \sum_{i=1}^{M} \sum_{y^t \in \mathcal{Y}_i^\epsilon} \Pr(Y^t = y^t, \theta = i) \tag{3.75}$$

$$\leq \frac{1}{C} \sum_{i=1}^{M} \sum_{y^t \in \mathcal{Y}_i^\epsilon} \mathsf{E}[U_i(t+1) - U_i(t) \,|\, Y^t = y^t, \theta = i] \Pr(Y^t = y^t, \theta = i)\,. \tag{3.76}$$

By Lemma (3), expression (3.76) is equal to the left side of inequality (3.68), which is bounded by (3.77) according to Lemma 6:

$$\frac{1}{C} \sum_{i=1}^{M} \Pr(\theta = i) \sum_{n=1}^{\infty} \mathsf{E}[U_i(T_n) - U_i(t_0^{(n)}) \,|\, \theta = i] \leq \left(1 + 2^{-C_2} \frac{1 - 2^{-NC_2}}{1 - 2^{-C_2}}\right) \frac{C_2}{C} - \frac{\mathbf{U}(Y^0)}{C}\,, \tag{3.77}$$

where $\mathbf{U}(Y^0)$ is the expected value of the log likelihood ratio of the true message according to the *a-priori* message distribution: $\mathbf{U}(Y^0) = \sum_{i \in \Omega} \rho_i(0) \log_2 \left(\frac{\rho_i(0)}{1 - \rho_i(0)}\right)$. Note that $\mathbf{U}(Y^0)$ is $-\log(M-1)$ for a uniform *a-priori* input distribution. Equations (3.75)-(3.77) yield the following bound on $\mathsf{E}[T]$:

$$\mathsf{E}[T] \leq 2^{-C_2} \frac{1 - 2^{-NC_2}}{1 - 2^{-C_2}} \frac{C_2}{C} + \frac{C_2 - \mathbf{U}(Y^0)}{C}\,. \tag{3.78}$$

The expectation $\mathsf{E}[\tau - T]$ is bounded via Lemma 5:

$$\mathsf{E}[\tau - T] \leq \left(N - 2^{-C_2} \frac{1 - 2^{-NC_2}}{1 - 2^{-C_2}}\right) \frac{C_2}{C_1}\,. \tag{3.79}$$

Note that (3.79) is not equality because Lemma 5 uses the "Gene aided" decoder that does not

terminate the transmission if any message j, other than the transmitted message θ, attains $U_j(t) \geq NC_2$. However, the actual process does end in error whenever such condition is met by any message.

A bound similar to (3.79) was first obtained by Yang *et al.* [YPA21], Section V. *F*, using a Markov Chain analysis. Yang's bound differs in that it includes in $\tau - T$ the time spent by the correct message in the communication phase, after a fall back event. This time is included in the self loop weight $\Delta_0 = 1 + \frac{C_2}{C} + \frac{C_2}{C}$, and later replaced by $\Delta_0 = 1 + \frac{C_2}{C} + \frac{\log_2(2q)}{qC}$, using the surrogate process analysis of Thm. 2 adapted to the SED rule. The analysis analysis by Yang *et al.* could also be used in (3.79), if the self loop is set to $\Delta_0 = 1$, as was done in [AGW23]. This is because all the time spend in the communication phase is already included in bound (3.78).

The upper bound on the expected stopping time $\mathsf{E}[\tau]$ is obtained by adding the bounds in equations (3.78) and (3.79), and then replacing N by its definition in equation (3.18):

$$\mathsf{E}[\tau] \leq \frac{\log_2(M-1) + C_2}{C} + \frac{C_2}{C_1}\left\lceil \frac{\log_2\left(\frac{1-\epsilon}{\epsilon}\right)}{C_2} \right\rceil + 2^{-C_2}\frac{1 - \frac{\epsilon}{1-\epsilon}2^{-C_2}}{1-2^{-C_2}}\left(\frac{C_2}{C} - \frac{C_2}{C_1}\right). \tag{3.80}$$

The proof of Thm. 1 is complete. □

3.12 Proof Thm. 2 and Thm. 3

Proof of Thm. 2. Suppose $U_i'(t)$ is a process that satisfies the constraints (3.22)-(3.26) in Thm. 1 and constraints (3.28)-(3.30) of Thm. 2 for some $B < C_2$. Because the constraints of Thm. 1 are satisfied, Lemmas 1-6 all hold for the process $U_i'(t)$. Then, $\mathsf{E}[T']$ can be bounded by the sum on the right side of Lemma 3, which is (3.52), but using the new process $U_i'(t)$. Then, to produce the desired result, it suffices to divide the new bound by C. The proof of Thm. 2 follows the procedure used in the proof of Lemma 6, but replacing $U_i(t)$ by $U_i'(t)$, up to the equation (3.328). Note that $U_i'(t_0^{(n)}) = U_i(t_0^{(n)})$, by the definition

of $U_i'(t)$. From equation (3.25), it follows that, for $n > 1$, the event $\{T^{(n)} > 0\}$ implies that $U_i(t_0^{(n)}) = U_i(T_{n-1}) - C_2$. Then, from equations (3.328) to (3.336), $U_i'(t_0^{(n)})$ can be replaced by $U_i(T_{n-1}) - C_2$. Then, from equation (3.336) the following inequality is obtained:

$$\sum_{n=1}^{\infty} \mathsf{E}[U_i'(T_n') - U_i'(t_0^{(n)}) \mid \theta = i] \leq -U_i'(0) + \sum_{n=1}^{\infty} \mathsf{E}[U_i'(T_n') - p_f(U_i(T_n) - C_2) \mid T^{(n)} > 0, \theta = i] p_f^{n-1}$$

From the definition of $U_i'(t)$, $U_i'(0)$ can be replaced by $U_i(0)$. Then, by the constraints of Thm. 2, $U_i'(T_n') - p_f(U_i(T_n) - C_2) \leq B$. This is derived from constraint (3.30): $U_i'(T_n') - \frac{p}{q}(U_i(T_n) - C_2) \leq B$ by replacing p_f by $\frac{p}{q}$. The replacement is possible because $p_f < \frac{p}{q}$, see (3.263), and $U_i(T_n) - C_2 < 0$ by constraint (3.23). Therefore, the expectation in the last sum can be replaced with B for an upper bound, to obtain:

$$\sum_{n=1}^{\infty} \mathsf{E}[U_i'(T_n') - U_i(t_0^{(n)}) \mid \theta = i] \leq -U_i(0) + \sum_{n=1}^{\infty} B p_f^{n-1} \tag{3.81}$$

$$= B + \frac{B p_f}{1 - p_f} - U_i(0) = B + 2^{-C_2} \frac{1 - 2^{-NC_2}}{1 - 2^{-C_2}} B - U_i(0)\,. \tag{3.82}$$

Then, the value in equation (3.82) replaces the inner sum in the left side of (3.77), to obtain:

$$\frac{1}{C} \sum_{i=1}^{M} \Pr(\theta = i) \sum_{n=1}^{\infty} \mathsf{E}[U_i'(T_n') - U_i'(t_0^{(n)}) \mid \theta = i]$$

$$\leq \frac{1}{C} \sum_{i=1}^{M} \Pr(\theta = i) \left(U_i(0) + B + 2^{-C_2} \frac{1 - 2^{-NC_2}}{1 - 2^{-C_2}} B \right)$$

$$= \frac{B}{C} \left(1 + 2^{-C_2} \frac{1 - 2^{-NC_2}}{1 - 2^{-C_2}} \right) - \frac{\mathsf{E}[U_i(0)]}{C}\,. \tag{3.83}$$

The proof is complete. □

Proof of Thm. 3. Note first, that if $\exists i \in \Omega$, with $U_i(t) \geq 0$, then constraint (3.32) is the same as the SED constraint (2.9) and therefore, the constraints (3.25) and (3.24) are satisfied as shown in [YPA21]. To prove Thm. 3, it suffices to show that constraints (3.22), (3.23) and

(3.26) are satisfied when $U_i(t) < 0 \, \forall i \in \Omega$. Start the proof by deriving expressions for $\mathsf{E}[U_i(t+1) - U_i(t) \mid Y^t, \theta = i]$ to find bounds in terms of the constraints of the theorem. The posterior probabilities $\rho_i(y^{t+1})$ are computed according to Bayes' Rule:

$$\rho_i(y^{t+1}) = \frac{\Pr(\theta = i, Y_{t+1} = y_{t+1} \mid Y^t)}{\Pr(Y_{t+1} = y_{t+1} \mid Y^t)} \,. \tag{3.84}$$

The top conditional probability in equation (3.84) can be split into $P(Y_{t+1} = y_{t+1} \mid \theta = i, Y^t = y^t) \Pr(\theta = i \mid y^t)$. Since the received history y^t fully characterizes the vector of posterior probabilities $\boldsymbol{\rho}_t \triangleq [\rho_1(y^t), \rho_2(y^t), \ldots, \rho_M(y^t)]$, and the new construction of S_0 and S_1, then the conditioning event $\{\theta = i\}$ sets the value of the encoding function $X_{t+1} = \mathrm{enc}(i, Y^t)$, via its definition: $\mathrm{enc}(i, Y^t) = \mathbb{1}_{i \in S_1}$. The first probability can be written as $\Pr(Y_{t+1} \mid \mathrm{enc}(i, Y^t))$, which reduces to q if $Y_{t+1} = \mathrm{enc}(i, Y^t)$ and to p if $Y_{t+1} \neq \mathrm{enc}(i, Y^t)$. The second probability $\Pr(\theta = i \mid Y^t = y^t)$ is just $\rho_i(y^t)$. The bottom conditional probability can be written as $\sum_{x_{t+1} \in \{0,1\}} \Pr(Y_{t+1} = y_{t+1} \mid X_{t+1}, Y^t) P(X_{t+1} = x_{t+1} \mid Y^t)$. By the channel memoryless property, the next output Y_{t+1} given the input X_{t+1} is independent of the past Y^t, that is: $\Pr(Y_{t+1} = y_{t+1} \mid X_{t+1}, Y^t) = \Pr(Y_{t+1} = y_{t+1} \mid X_{t+1})$. Since $\Pr(X_{t+1} = x_{t+1} \mid Y^t) = \Pr(\theta \in S_{x_{t+1}})$ which is given by $\sum_{i \in S_{x_{t+1}}} \rho_i(y^t)$, then:

$$\rho_i(t+1) = \frac{\Pr(Y_{t+1} \mid i) \rho_i(y^t)}{\sum_{j \in \Omega} \Pr(Y_{t+1} \mid j) \rho_j(y^t)} = \frac{\Pr(Y_{t+1} \mid i) \rho_i(y^t)}{q \sum_{j \in S_{y_{t+1}}} \rho_j(y^t) + p \sum_{j \in \Omega \setminus S_{y_{t+1}}} \rho_j(y^t)} \,. \tag{3.85}$$

For $\{i = \theta\}$ the encoding function $X_{t+1} = \mathbb{1}_{\theta \in S_1}$ dictates that $X_{t+1} = \mathbb{1}_{i \in S_1}$. Thus $\Pr(Y_{t+1} = \mathbb{1}_{i \in S_1} \mid i = \theta) = \Pr(Y_{t+1} = X_{t+1}) = q$, and $\Pr(Y_{t+1} = \mathbb{1}_{i \notin S_1} \mid i = \theta) = \Pr(Y_{t+1} = X_{t+1} \oplus 1) = p$. Let $P_0 = \sum_{j \in S_0} \rho_j(y^t)$ and $P_1 = \sum_{j \in S_1} \rho_j(y^t)$, and let $\Delta \triangleq P_0 - P_1$, so that $P_0 = \frac{1}{2} + \frac{\Delta}{2}$ and $P_1 = \frac{1}{2} - \frac{\Delta}{2}$. The value of $U_i(t+1)$ for each $Y_{t+1} \in \{0, 1\}$ can be obtained from equation

3.85. Assume first that $i \in S_0$ to obtain the value of $E[U_i(t+1) - U_i(t) \mid Y^t = y^t, \theta = i]$.

$$\begin{aligned}\mathsf{E}[U_i(t+1) \mid Y^t = y^t, \theta = i] &= q \log_2 \frac{\frac{\rho_i(y^t)q}{P_0 q + P_1 p}}{1 - \frac{\rho_i(y^t)q}{P_0 q + P_1 p}} + p \log_2 \frac{\frac{\rho_i(y^t)p}{P_0 p + P_1 q}}{1 - \frac{\rho_i(y^t)p}{P_0 p + P_1 q}} \\ &= q \log_2 \frac{\rho_i(y^t)q}{\frac{1}{2} + \frac{\Delta(q-p)}{2} - \rho_i(y^t)q} + p \log_2 \frac{\rho_i(y^t)p}{\frac{1}{2} - \frac{\Delta(q-p)}{2} - \rho_i(y^t)p}.\end{aligned}$$

For $i \in S_1$ the only difference is the sign of the term with Δ. Let $\iota_i = \mathbb{1}_{i \in S_0} - \mathbb{1}_{i \in S_1}$, that is 1 if $i \in S_0$ and -1 if $i \in S_1$ and add a coefficient ι_i to each Δ for a general expression. Multiply by 2 both terms of the fraction inside the logarithm and expand it to obtain:

$$\mathsf{E}[U_i(t+1) - U_i(t) \mid Y^t, \theta = i] = \log_2(\rho_i(y^t)) \tag{3.86}$$

$$+ q\big(\log_2(2q) - \log_2\left(1 - \rho_i(y^t) + (q-p)(\iota_i\Delta - \rho_i(y^t))\right)\big)$$

$$+ p\left(\log_2(2p) - \log_2\left(1 - \rho_i(y^t) - (q-p)(\iota_i\Delta - \rho_i(y^t))\right)\right) \tag{3.87}$$

$$= q\left(\log_2(2q) - \log_2\left(1 + (q-p)\frac{\iota_i\Delta - \rho_i(y^t)}{1 - \rho_i(y^t)}\right)\right)$$

$$+ p\left(\log_2(2p) - \log_2\left(1 - (q-p)\frac{\iota_i\Delta - \rho_i(y^t)}{1 - \rho_i(y^t)}\right)\right) \tag{3.88}$$

$$\geq C - \log_2\left(1 + (q-p)^2 \frac{\iota_i\Delta - \rho_i(y^t)}{1 - \rho_i(y^t)}\right). \tag{3.89}$$

Now subtract the term $\log_2(1 - \rho_i(y^t))$, and add it back as a factor in the logarithm, to recover $U_i(t)$ from $\log_2(\rho_i(y^t))$. Note that $2\rho_i(y^t)q = \rho_i(y^t) + (q-p)\rho_i(y^t)$ and $2\rho_i(y^t)p = \rho_i(y^t) - (q-p)\rho_i(y^t)$. And also note that $q\log_2(2q) + p\log_2(2p) = C$.

The logarithm $\log_2(1 - \rho_i(y^t))$ from (3.87) is split into $p\log_2(1 - \rho_i(y^t)) + q\log_2(1 - \rho_i(y^t))$, and $1 - \rho_i(y^t)$ divides the arguments of the logarithms in (3.88). Equation (3.89) follows

from applying Jensen's inequality to the convex function $-\log_2(\cdot)$. Then:

$$\sum_{i=1}^{M} \mathsf{E}[U_i(t+1)-U_i(t)\,|\,Y^t,\theta=i]\rho_i(Y^t) \geq C-\sum_{i=1}^{M}\rho_i(y^t)\log_2\left(1+(q-p)^2\frac{\iota_i\Delta-\rho_i(y^t)}{1-\rho_i(y^t)}\right) \tag{3.90}$$

$$= C-\sum_{i\in S_0}\rho_i(y^t)\log_2\left(1+(q-p)^2\frac{\Delta-\rho_i(y^t)}{1-\rho_i(y^t)}\right) \tag{3.91}$$

$$-\sum_{i\in S_1}\rho_i(y^t)\log_2\left(1-(q-p)^2\frac{\Delta+\rho_i(y^t)}{1-\rho_i(y^t)}\right). \tag{3.92}$$

By the SEAD constraints, given by equations (3.33) and (3.32), if $i \in S_0$, then $\Delta \leq \rho_{\min} \leq \rho_i(y^t)$. For the case where $\Delta \geq 0$, then $i \in S_0 \implies \Delta - \rho_i(y^t) \leq 0$ and $-\Delta - \rho_i(y^t) < 0$. Then the arguments of the logarithms in (3.92) are both less than 1 for every i. This suffices to show that the constraints (3.26) and (3.22) are satisfied when $P_0 \geq P_1$ for the case that $\Delta \geq 0$.

It remains to prove that constraints (3.26) and (3.22) hold in the case where $P_1 > P_0$, or equivalently $\Delta < 0$. Let $\alpha = -\Delta > 0$, and note that since $0 < \alpha < 1$, then:

$$\frac{\alpha}{1-\rho_{\min}} \geq \alpha = \alpha\frac{1-\rho_i(y^t)}{1-\rho_i(y^t)} \geq \frac{\alpha-\rho_i(y^t)}{1-\rho_i(y^t)}, \tag{3.93}$$

and $\rho_i \geq \rho_{\min} \implies \alpha+\rho_i \geq \alpha+\rho_{\min}$ and $1-\rho_i < 1-\rho_{\min}$, therefore:

$$\log_2\left(1-(q-p)^2\frac{\alpha+\rho_i(y^t)}{1-\rho_i(y^t)}\right) \leq \log_2\left(1-(q-p)^2\frac{\alpha+\rho_{\min}}{1-\rho_{\min}}\right)$$
$$\log_2\left(1+(q-p)^2\frac{\alpha-\rho_i(y^t)}{1-\rho_i(y^t)}\right) \leq \log_2\left(1+(q-p)^2\frac{\alpha}{1-\rho_{\min}}\right).$$

Since this holds for all $i = 1, \ldots, M$, then:

$$\sum_{i=1}^{M} \mathsf{E}[U_i(t+1) - U_i(t) \mid Y^t = y^t, \theta = i]\rho_i(y^t) \geq C \tag{3.94}$$

$$- P_0 \log_2\left(1-(q-p)^2\frac{\alpha+\rho_{\min}}{1-\rho_{\min}}\right) - P_1 \log_2\left(1+\alpha\frac{(q-p)^2}{1-\rho_{\min}}\right) \tag{3.95}$$

$$\geq C - \log_2\left(1 - \frac{(q-p)^2}{1-\rho_{\min}}[P_0(\alpha+\rho_{\min}) - P_1\alpha]\right). \tag{3.96}$$

To satisfy constraint (3.26) the logarithm term in (3.96) needs to be non-negative. This only requires that $-\Delta^2 + P_0\rho_{\min} > 0$. Since $P_0 - P_1 = \Delta$, then $P_0(\alpha+\rho_{\min}) - P_1\alpha = (P_0 - P_1)\alpha + P_0\rho_{\min} = -\Delta^2 + P_0\rho_{\min}$. To satisfy constraint (3.26) it suffices that $-\Delta^2 + P_0\rho_{\min} > 0$, which is equivalent to:

$$\Delta^2 \leq P_0\rho_{\min}\,. \tag{3.97}$$

The SEAD constraints, equations (3.33) and (3.32), guarantees that $\Delta^2 \leq \rho_{\min}^2$. Since $P_0 \geq \min_{i \in S_0} \rho_i(y^t) = \rho_{\min}$, then $\Delta^2 \leq \rho_{\min}^2 \leq P_0\rho_{\min}$, which satisfies inequality (3.97). Then, the SEAD constraints guarantee that constraint (3.26) is satisfied, and only restricts the absolute difference between P_0 and P_1.

To prove that constraint (3.22) is satisfied, note that equation (3.33) of the SEAD constraints guarantees that if $\rho_j(t) \leq \frac{1}{2}\ \forall j = 1, \ldots, M$, then $|\Delta| \leq \frac{1}{3}$. Starting from equation (3.89) note that the worst case scenario is when $\iota_i\Delta = \frac{1}{3}$. From (3.98) to (3.99), use (3.93)

with $\alpha = \frac{1}{3}$ to obtain:

$$\begin{aligned}
\mathsf{E}[U_i(t+1) - U_i(t) \mid Y^t, \theta = i] \geq & C - \log_2 \left(1 + (q-p)^2 \frac{\iota_i \Delta - \rho_i(y^t)}{1 - \rho_i(y^t)}\right) && (3.98)\\
\geq & C - \log_2 \left(1 + \frac{(q-p)^2}{3}\right) && (3.99)\\
\geq & C - \frac{(q-p)^2}{3} && (3.100)\\
= & C - \frac{(q-p)^2}{2\ln(2)} + \frac{3 - 2\ln(2)}{6\ln(2)}(q-p)^2 && (3.101)\\
\geq & \frac{3 - 2\ln(2)}{6\ln(2)}(q-p)^2 > \frac{(q-p)^2}{3}\,. && (3.102)
\end{aligned}$$

To transition from (3.101) to (3.102), need to show that $2\ln(2)C \geq (q-p)^2$. To show this, it suffices to find a small constant a that makes $aC - (q-p)^2$, the difference between two convex functions, also convex. Take second derivatives $\frac{d^2}{dp^2} aC = \frac{1}{\ln(2)} \frac{a}{pq}$ and $\frac{d^2}{dp^2}(q-p)^2 = 8$, and subtract them. The constant a is found by noting that $pq \leq \frac{1}{4}$.

The SEAD constraints guarantee that both sets, S_0 and S_1 are non-empty. Since the maximum absolute value difference $\mid U_i(t+1) - U_i(t) \mid$ is C_2, then, constraint (3.23) is satisfied, see the proof of *Claim 2*.

For the proof of existence of a process $U_i'(t)$, with $B = \frac{1}{q}\log_2(2q)$, see Sec. 3.19. □

3.13 Proof of Converse Thm. 4

Proof of Thm. 4. By Lemma 1, equation (3.46) the expectation $\mathsf{E}[\tau]$ is given by:

$$\mathsf{E}[\tau] = \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}^\epsilon} \Pr(Y^t = y^t \mid \theta = i) \tag{3.103}$$

This expectation can be separated into sums over the sets $\mathcal{A}_\epsilon$ defined in Lemma 2, the set difference $\mathcal{Y}_i^\epsilon \setminus \mathcal{A}_\epsilon$, and the set difference $\mathcal{Y}^\epsilon \setminus \mathcal{Y}_i^\epsilon$, see Lemma 1.

$$\mathsf{E}[\tau] = \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}^\epsilon} \Pr(Y^t = y^t \mid \theta = i) \tag{3.104}$$

$$= \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{A}_\epsilon} \Pr(Y^t = y^t \mid \theta = i) \tag{3.105}$$

$$+ \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}_i^\epsilon \setminus \mathcal{A}_\epsilon} \Pr(Y^t = y^t \mid \theta = i) \tag{3.106}$$

$$+ \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}^\epsilon \setminus \mathcal{Y}_i^\epsilon} \Pr(Y^t = y^t \mid \theta = i) \tag{3.107}$$

Let τ_0, τ_1, τ_2 be each of these sums, and also define the sum S by:

$$\mathcal{Y}_i^t \triangleq \{y^t \in \{0,1\}^t \mid \rho_i(y_1^s) < \frac{1}{2} \, \forall s = 1, 2, \ldots, t\} \tag{3.108}$$

$$\mathcal{Y}_i \triangleq \cup_{t=0}^{\infty} \mathcal{Y}_i^t \tag{3.109}$$

$$\tau_0 = \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{A}_\epsilon} \Pr(Y^t = y^t \mid \theta = i) \tag{3.110}$$

$$\tau_1 = \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}_i^\epsilon \setminus \mathcal{A}_\epsilon} \Pr(Y^t = y^t \mid \theta = i) \tag{3.111}$$

$$\tau_2 = \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}^\epsilon \setminus \mathcal{Y}_i^\epsilon} \tag{3.112}$$

$$S = \frac{1}{C} \sum_{i=1}^{M} \sum_{y^t \in \mathcal{Y}_i^\epsilon \setminus \mathcal{A}_\epsilon} \mathsf{E}[\log_2(\rho_i(y^{t+1})) - \log_2(\rho_i(y^t)) | \theta = i, Y^t = y^t] \Pr(Y^t = y^t, \theta = i) \, . \tag{3.113}$$

Then:

$$\mathsf{E}[\tau] = \tau_0 + \tau_1 + \tau_2 = (\tau_0 + S) + (\tau_1 - S + \tau_2) \tag{3.114}$$

The proof consists of finding bounds on $\tau_0 + S$ and on $\tau_1 - S + \tau_2$. From Lemma 2, the time τ_0 is lower bounded by:

$$\tau_0 = \sum_{i=1}^{M} \Pr(\theta{=}i) \sum_{y^t \in \mathcal{A}_\epsilon} \Pr(Y^t = y^t \mid \theta{=}i) \tag{3.115}$$

$$\geq \sum_{i=1}^{M} \frac{\Pr(\theta{=}i)}{C} \sum_{y^t \in \mathcal{A}_\epsilon} \mathsf{E}[\log_2(\rho_i(y^{t+1})) - \log_2(\rho_i(y^t)) | Y^t{=}y^t, \theta{=}i] \Pr(Y^t{=}y^t \mid \theta{=}i) \tag{3.116}$$

Then, the sum $\tau_0 + S$ is upper bounded by:

$$\tau_0 + S \geq \frac{1}{C} \sum_{i=1}^{M} \Pr(\theta{=}i) \sum_{y^t \in \mathcal{A}_\epsilon} \mathsf{E}[\log_2 \left(\frac{\rho_i(y^{t+1}))}{\rho_i(y^t)} \right) | Y^t{=}y^t, \theta{=}i] \Pr(Y^t{=}y^t \mid \theta{=}i) \tag{3.117}$$

$$+ \frac{1}{C} \sum_{i=1}^{M} \Pr(\theta{=}i) \sum_{y^t \in \mathcal{Y}_i^\epsilon \setminus \mathcal{A}_\epsilon} \mathsf{E}[\log_2 \left(\frac{\rho_i(y^{t+1}))}{\rho_i(y^t)} \right) | Y^t{=}y^t, \theta{=}i] \Pr(Y^t{=}y^t \mid \theta{=}i) \tag{3.118}$$

$$= \frac{1}{C} \sum_{i=1}^{M} \Pr(\theta{=}i) \sum_{y^t \in \mathcal{Y}_i^\epsilon} \mathsf{E}[\log_2 \left(\frac{\rho_i(y^{t+1}))}{\rho_i(y^t)} \right) | Y^t{=}y^t, \theta{=}i] \Pr(Y^t{=}y^t \mid \theta{=}i) \,. \tag{3.119}$$

By Lemma 3, equation (3.53), the inner sum in (3.119) is given by:

$$\sum_{y^t \in \mathcal{Y}_i^\epsilon} \mathsf{E}[\log_2 \left(\frac{\rho_i(y^{t+1})}{\rho_i(y^t)} \right) | Y^t = y^t, \theta = i] \Pr(Y^t{=}y^t \,|\, \theta{=}i) = \mathsf{E}[\log_2 \left(\frac{\rho_i(y^T)}{\rho_i(0)} \right) \mid \theta{=}i] \,. \tag{3.120}$$

Then:

$$\tau_0 + S \geq \frac{1}{C} \sum_{i=1}^{M} \Pr(\theta{=}i) \mathsf{E}[\log_2(\rho_i(y^T)) - \log_2(\rho_i(0)) \mid \theta{=}i] \tag{3.121}$$

$$= \frac{1}{C} \sum_{i=1}^{M} (-1 - \log_2(\rho_i(0)) \Pr(\theta{=}i) + \frac{1}{C} \sum_{i=1}^{M} \mathsf{E}[\log_2(\rho_i(y^T)) + 1 | \theta{=}i] \Pr(\theta{=}i) \,. \tag{3.122}$$

Where the expectation in (3.121) is split into two in (3.122) parts and a term -1 is added to the first expectation and subtracted in the other. The first expectation is a constant given the initial distribution $\rho_i(0)$, and only the second expectation depends on the random crossing point at time $t = T$, compactly described by $\mathsf{E}[\log_2(\rho_\theta(y^T))] + 1$. Note that $\rho_i(y^T) \in [\frac{1}{2}, q)$ when a message i enters the confirmation phase.

If the transmitted message θ reaches its confirmation phase at any point in the transmission (where $\{\rho_i(y^T) \geq \frac{1}{2}, i = \theta\}$), then $\log_2(\rho_i(y^T)) \in [-1, \log_2(q))$, for $i = \theta$. However, there is a non-zero probability that the process terminates with a wrong estimate $\hat{\theta} \neq \theta$, without the event $\{\rho_i(y^T) \geq \frac{1}{2}, i = \theta\}$ ever occurring. Let $\mathcal{H}_0$ be the event where the first message i that crosses $\rho_i(y^T) \geq \frac{1}{2}$ is the transmitted message θ, given by $\mathcal{H}_0 \triangleq \{t = T, \rho_i(y^T) \geq 0, i = \theta\}$, and let $\mathcal{H}_1$ be the complement of $\mathcal{H}_0$. Using the law of total probability, the expected time τ_0 defined in (3.110) could be expressed as:

$$\tau_0 = \sum_{i=1}^{M} \Pr(\theta = i) \Pr(\mathcal{H}_0) \sum_{y^t \in \mathcal{A}_\epsilon} \Pr(Y^t = y^t \mid \theta = i, \mathcal{H}_0) \tag{3.123}$$

$$+ \sum_{i=1}^{M} \Pr(\theta = i) \Pr(\mathcal{H}_1) \sum_{y^t \in \mathcal{A}_\epsilon} \Pr(Y^t = y^t \mid \theta = i, \mathcal{H}_1)\,. \tag{3.124}$$

Since $\Pr(\hat{\theta} \neq \theta) \leq \epsilon$, then $\Pr\{\mathcal{H}_1, \hat{\theta} \neq \theta\} < \epsilon$, where the strict inequality is because $\Pr\{\mathcal{H}_0, \hat{\theta} \neq \theta\} > 0$. Then, $\Pr\{\mathcal{H}_1, \hat{\theta} = \theta\} \geq \Pr(\mathcal{H}_1) - \epsilon$. By the definition of the posterior probabilities $\rho_i(y^t)$ at time $t = T$, if $\rho_i(y^T) \geq \frac{1}{2}$, then $\Pr(\theta = i \mid Y^T = y^T) = \rho_i(y^T) \geq \frac{1}{2}$. Then $\Pr(\mathcal{H}_1) \leq \frac{1}{2}$. This allows a bound on τ that excludes the cases where the transmitted message θ never makes it to the confirmation phase, with probability bounded by ϵ out of event $\mathcal{H}_1$, at the expense of also excluding some events where θ does reach the confirmation phase from event $\mathcal{H}_0$. The same fraction $\frac{\mathcal{H}_1 - \epsilon}{\mathcal{H}_1}$ excluded from event $\mathcal{H}_1$ must also be excluded from event $\mathcal{H}_0$. Then, the expression for $\tau_0 + S$ in (3.122) needs to be multiplied by a factor

r_h given by:

$$r_h \triangleq (\Pr(\mathcal{H}_1) - \epsilon) + \Pr(\mathcal{H}_0)\frac{\Pr(\mathcal{H}_1) - \epsilon}{\Pr(\mathcal{H}_1)}\,. \tag{3.125}$$

The expressions (3.115)-(3.122) may be split into two, each with conditioning events $\mathcal{H}_0$ and $\mathcal{H}_1$ and multiplied by the event probabilities $\Pr(\mathcal{H}_0)$ and $\Pr(\mathcal{H}_1)$ as was done with the expression for τ_0 in (3.123) and (3.124). To exclude from Eq. (3.124) the event where the transmitted message never reaches the confirmation phase (given by $\{\forall t \leq \tau : i = \theta, \rho_i(y^t) < \frac{1}{2}\}$), both (3.124) with $\mathcal{H}_0$ and (3.124) must be multiplied by the factor r_h. Then $\mathsf{E}[\log_2(\rho_\theta(y^T))] + 1 \in [0, 1 + \log_2(q))$ when restricted to such events, since $\rho_i(y^T) \in [\frac{1}{2}, q)$ when message i enters its confirmation phase. This leads to the following bound on $\tau_0 + S$:

$$\tau_0 + S \geq r_h\frac{\mathsf{E}[\rho_\theta(0) - 1]}{C} + r_h\frac{\mathsf{E}[\log_2(\rho_\theta(y^T))] + 1}{C}. \tag{3.126}$$

For uniform input distribution over $\{0,1\}^K$ the value of $\rho_i(0)$ is exactly $-K$ and bound (3.126) becomes:

$$\tau_0 + S \geq r_h\frac{K - 1}{C} + r_h\frac{\mathsf{E}[\log_2(\rho_\theta(y^T))] + 1}{C}. \tag{3.127}$$

The lower bound on $\tau_0 + S$ of (3.127) can be further refined by including some part of the communication phase for the other $1 - r_h$ part of the transmission. Note that the communication phase stops at a time T when a candidate message i attains $\rho_i(y^t) \geq \frac{1}{2}$. The largest possible step size that the process $\log_2(\rho_i(Y^t))$ may take, under optimal equal partitioning, is $\log_2(2q)$. This step size is taken by a message i whenever $i \in S_{y_t}$. Then, the minimum time for any candidate i to reach the confirmation phase, under equal partitioning is given by:

$$\min\{T\} = \frac{-1 - \log_2(\rho_i(0))}{\log_2(2q)}\,. \tag{3.128}$$

Expression (3.128) for uniform input distribution over $\{0,1\}^K$ becomes $\frac{K-1}{\log_2(2q)}$ and the bound on $\tau_0 + S$ becomes:

$$\tau_0 + S \geq r_h \frac{K-1}{C} + r_h \frac{\mathsf{E}[\log_2(\rho_\theta(y^T))] + 1}{C} + (1 - r_h) \frac{K-1}{\log_2(2q)}. \tag{3.129}$$

The next step consist of finding an upper bound on S, which needs to be subtracted from lower bound in the expected decoding time $\mathsf{E}[\tau]$. Let ρ_n be the posterior at which a wrong message enters the confirmation phase for the n-th consecutive time, before an initial correct confirmation phase, and let m denote the number of times such event occurs. Since ρ_n is the posterior of a message that has entered confirmation, then, the probability that it is not the message θ is exactly $1 - \rho_n$.

By Lemma 7, the message in the wrong confirmation phase returns to a posterior below $\frac{1}{2}$ *a.s.*. Then n consecutive wrong confirmation rounds happen with probability $\Pr(m \geq n) = \prod_{r=1}^{n}(1 - \rho_r)$. Using Lemma 8, the value of S can be expressed by:

$$S = \frac{1}{C} \sum_{i=1}^{M} \Pr(\theta = i) \sum_{n=1}^{\infty} V_q(\rho_n) \Pr(m \geq n) \tag{3.130}$$

$$= \frac{1}{C} \sum_{i=1}^{M} \Pr(\theta = i) \sum_{n=1}^{\infty} \prod_{r=1}^{n} (1 - \rho_r) \log_2 \frac{q}{q(1-\rho_n) + p\rho_n}\,. \tag{3.131}$$

Each time the n consecutive wrong confirmation phase happens, the term $\frac{1}{C} V_q(\rho_n)$ is added to S, where:

$$V_q(\rho_n) = \log_2 \frac{q}{q(1-\rho_n) + p\rho_n} \tag{3.132}$$

To obtain the time τ_0, the therm S needs to be replaced by the actual time spent in the wrong confirmation phase, which also requires that the process has not already ended in wrong decoding. The expected wrong confirmation time, from equation (3.65) of of Lemma

5, is given by:

$$\mathsf{E}[S^{bad}] = \frac{C_2}{C_1}\left(1 - 2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right) \tag{3.133}$$

Let $W_q(y^t)$ be the boosting update factor at time t, and let $W_p(y^t)$ be the attenuating update factor. That is, $W_q(y^t)$ applies to items in $S_{y_{t+1}}$, and $W_p(y^t)$ to items in set $S_{1\oplus y_{t+1}}$. Define the set $\mathcal{E}_\epsilon$ as the set of y^t from $\mathcal{A}_\epsilon$, for which at least one item can enter confirmation when boosted by $W_q(y^t) = 2q$, which happens with probability q if the $\theta \in S_{y_{t+1}}$, along with the item, and p otherwise. The set $\mathcal{E}_\epsilon$ is compactly defined by:

$$\mathcal{E}_\epsilon \triangleq \{y^t \in \mathcal{A}_\epsilon : \exists i : \rho_i(y^t)W_q(y^t) \geq \tfrac{1}{2}\} \tag{3.134}$$

Note that the second term in (3.122) is:

$$\frac{1}{C}\sum_{i=1}^{M}\mathsf{E}[\log_2(\rho_i(y^T)) + 1|\theta = i]\Pr(\theta = i) \tag{3.135}$$

$$= \frac{1}{C}\sum_{i=1}^{M}\Pr(\theta = i)\sum_{y^T\in\mathcal{E}^\epsilon}\log_2(2\rho_i(y^T))\Pr(Y^t = y^T \mid \theta = i) \tag{3.136}$$

$$= \frac{1}{C}\sum_{i=1}^{M}\Pr(\theta = i)\sum_{y^T\in\mathcal{E}^\epsilon}\Pr(Y^t = y^T)\log_2(2\rho_i(y^T))\rho_i(y^T) \tag{3.137}$$

Thus, a higher crossing probability ρ_n, which reduces the probability that the crossing message is not θ, also increases the term $\tau_0 + S$ for each $\{Y^t = y^T\}$ event, by exactly:

$$\frac{1}{C}\rho_n\log_2(2\rho_n) \tag{3.138}$$

At the same time, the message crossing into the confirmation phase with posterior ρ_n has

probability $1-\rho_n$ of being wrong, which increases S by a term:

$$\frac{1}{C}(1-\rho_n)\log_2\frac{q}{q(1-\rho_n)+p\rho_n} \tag{3.139}$$

scaled by the probability $\Pr(m \geq n)$.

The time τ_0 includes the expected crossing posterior $\mathsf{E}[\log_2(2\rho_i(y^T))|\theta=i]$ for each i, see (3.122). The next steps consist of lower bounding jointly the terms $(\tau_0+S)+(\tau_1-S)$, where These terms are given by the expectation of:

$$\tau_0+\tau_1 = \frac{\mathsf{E}[\rho_\theta(0)-1]}{C} + \frac{1}{C}\sum_{i=1}^{M}\Pr(\theta=i)\sum_{n=1}^{\infty}\prod_{r=1}^{n}(1-\rho_r)f(\rho_n) \tag{3.140}$$

$$f(\rho_n) \triangleq \left(\rho_n\log_2(2\rho_n) - (1-\rho_n)\log_2\frac{q}{q(1-\rho_n)+p\rho_n} + p_r^n(1-\rho_n)\mathsf{E}[S^{bad}]\right). \tag{3.141}$$

The last term $\mathsf{E}[S^{bad}]$ has an additional factor p_r^n, to account for the recovery probability defined in Lemma 4, since a subsequent wrong confirmation happens only if all previous ones end in recovery, and not in wrong decoding. The overall expectation can be lower bounded, using linearity of expectations, by lower bounding each term in (3.141). For each wrong confirmation event, it suffices to minimize $f(\rho_n)$. Thus, the single value $\rho_n \in [\frac{1}{2}, q)$ minimizes $f(\rho_n)$ for every $n = 1, 2, \ldots$. The next claim analyzes this minimization.

Claim 4: The value of ρ_n that minimizes $f(\rho_n)$ is $\rho_n = \frac{1}{2}$. *Recall from Lemma 5 that the expected duration of the wrong confirmation phase is:*

$$E[S_n^{bad}] = \frac{C_2}{C_1}\left(1 - 2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right) \tag{3.142}$$

Note that the factor p_r^n, the probability of n consecutive recoveries, strictly decreases the "weight" of the term with $E[S_n^{bad}]$ in $f(\rho_n)$. Since $E[S_n^{bad}]$ is positive, as the wrong confirmation time is at least one time unit, replacing p_r^n by 1, strictly increases $f(\rho_n)$. With this

replacement, let $f(\rho_n)$ *be:*

$$f(\rho_n) = -(1-\rho_n)\frac{1}{C}\log_2\frac{q}{q(1-\rho_n)+p\rho_n} + \frac{\rho_n}{C}\log_2(2\rho_n) \tag{3.143}$$

$$+ (1-\rho_n)\frac{C_2}{C_1}\left(1 - 2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right) \tag{3.144}$$

To show that $\tau_0+\tau_1$ *is minimized if* $\rho_n = \frac{1}{2}$ *for all* $n = 1, 2, \ldots,$ *it suffices to show that* $f(\rho_n)$, $\rho_n \in [\frac{1}{2}, 1-q)$ *or* $f(1-x)$, $x \in (1-q, \frac{1}{2}]$ *is minimized at* $1-x = \frac{1}{2}$. *Note that the only difference between the exact correction term and* $f(\rho_n)$, *is that in* $f(\rho_n)$, *a factor* $P_n \in (0,1)$, *of a positive and linear term, was set to* 1 *in* $f(\rho_n)$. *If the relaxed* $f(\rho_n)$ *is still minimized at the largest possible value for that term, then the actual value is also minimized at the same value of* ρ_n.

The proof is in 3.13

From claim 4, the crossing value $\rho_i(y^T) = \frac{1}{2}$, and $\log_2(\rho_i(y^T)) = -1$. If $\rho_i(y^T) = \frac{1}{2}$, then $\Pr(\mathcal{H}_1) = \frac{1}{2}$ and the value of r_h in (3.125) is $r_h = 1-2\epsilon$. Then, a lower bound on $\tau_0+\tau_1$ is given by jointly bounding $\tau_0 + S + (\tau_1 - S)$ where a bound on $\tau_0 + S$ for uniform input distribution over $\{0,1\}^K$, is computed from (3.129) with $r_h = 1-2\epsilon$ as follows:

$$\tau_0 + S \geq (1-2\epsilon)\frac{K-1}{C} + (1-2\epsilon)\frac{\mathsf{E}[\log_2(\rho_\theta(y^T))]+1}{C} + 2\epsilon\frac{K-1}{\log_2(2q)} \tag{3.145}$$

$$\geq (1-2\epsilon)\frac{K-1}{C} + (1-2\epsilon)\frac{\mathsf{E}[-1]+1}{C} + 2\epsilon\frac{K-1}{\log_2(2q)} \tag{3.146}$$

$$= (1-2\epsilon)\frac{K-1}{C} + 2\epsilon\frac{K-1}{\log_2(2q)}. \tag{3.147}$$

Where $E[\log_2(\rho_\theta(0)]$ is replaced by $-K$ since $\rho_i(0) = -K\ \forall i \in \{0,1\}^K$.

With the crossing value of $\rho_i(y^T) = \frac{1}{2}$ from claim 4, n consecutive bad confirmation

phases happen with probability $P_n = p_r^n(\frac{1}{2})^n$:

$$P_n = p_r^n \prod_{j=1}^{n}(1-\tfrac{1}{2})^j = p_r^n(\tfrac{1}{2})^n \tag{3.148}$$

and each $V_q(\rho_n)$ is exactly $\frac{1}{C}\log_2(2q)$. The value of S from the joint minimization is:

$$S = \frac{1}{C}\log_2(2q)\sum_{n=1}^{\infty}2^{-n} = \frac{1}{C}\log_2(2q) \tag{3.149}$$

Finally, the expected number of bad confirmation phases from the joint minimization is given by the sum of the first one with probability $\frac{1}{2}$, and every n-th subsequent one with probability $\frac{1}{2}p_r^n 2^{-n}$. Then the time τ_1 is given by the expected number of wrong confirmation rounds and their expected duration $\mathsf{E}[S_n]$, see Lemma 7, given by:

$$\begin{aligned}
\tau_1 &= \frac{1}{2}E[S_n]\sum_{n=0}^{\infty}2^{-n}p_r^n && (3.150)\\
&= \frac{1}{2}E[S_n]\sum_{i=0}^{\infty}\left(\frac{1}{2}\frac{1-2^{-NC_2}}{1-2^{-(N+1)C_2}}\right)^i && (3.151)\\
&= \frac{1}{2}E[S_n]\frac{1}{1-\frac{1}{2}\frac{1-2^{-NC_2}}{1-2^{-(N+1)C_2}}} && (3.152)\\
&= \frac{1-2^{-(N+1)C_2}}{2(1-2^{-(N+1)C_2})-(1-2^{-NC_2})}E[S_n] && (3.153)\\
&= \frac{1-2^{-(N+1)C_2}}{2(1-2^{-(N+1)C_2})-(1-2^{-NC_2})}\frac{C_2}{C_1}\left(1-2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right) && (3.154)
\end{aligned}$$

The correct confirmation phase happens with the exact same probability, that is, the first crossing probability $\rho_0 = \frac{1}{2}$, and every subsequent nth one with probability $2^{-(n+1)}p_r^n$. That is, after every recovery and subsequent crossing with $\rho_n(y^T) = \frac{1}{2}$, the probability of being correct is $\rho_n(y^T)$. To lower bound τ_2, suppose that at every fall back, a new message is in confirmation, (this maximizes the forward steps of $\rho_\theta(y^t)$). Suppose further that the process becomes a two way race, that is, $S_0 = \rho_\theta(y^t)$, even after a fall back, and a bad decoding

is declared if $1-\rho_i(y^t)$ reaches $1-\epsilon$. Clearly this could happen and still have no single item across, and the singleton choice $S_0 = \rho_\theta(y^t)$ maximizes the step size for $\rho_\theta(y^t)$. This results in a two way race, that can be modeled by a Markov Chain with $2N+1$ states $-N, -N+1, \ldots, -1, 0, 1, \ldots, N-1, N$, where $U_\theta(t)$ starts at state 0. The expected duration of such confirmation phase can be obtained using Corollary 2 by:

$$\mathsf{E}[S_{2N}] = \frac{C_2}{C_1}\left(N - 2^{-NC_2}\frac{1-2^{-NC_2}}{1-2^{-2NC_2}}2N\right) \tag{3.155}$$

Then, τ_2 from the joint minimization is given by:

$$\tau_2 = \frac{1-2^{-(N+1)C_2}}{2(1-2^{-(N+1)C_2}) - (1-2^{-NC_2})}\mathsf{E}[S_{2N}] \tag{3.156}$$

$$= \frac{1-2^{-(N+1)C_2}}{2(1-2^{-(N+1)C_2}) - (1-2^{-NC_2})}\frac{C_2}{C_1}\left(N - 2^{-NC_2}\frac{1-2^{-NC_2}}{1-2^{-2NC_2}}2N\right) \tag{3.157}$$

The bound τ_B is computed by adding the three jointly minimized terms $(\tau_0+S)+(\tau_1-S)+\tau_2$, and is given by:

$$\begin{aligned}\tau_B \triangleq\ & (1-2\epsilon)\frac{K-1}{C} + 2\epsilon\frac{K-1}{\log_2(2q)} - (1-2\epsilon)\frac{\log_2(2q)}{C} \\ & + \frac{1-2^{-(N+1)C_2}}{2(1-2^{-(N+1)C_2}) - (1-2^{-NC_2})}\frac{C_2}{C_1}\left(1 - 2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right) \\ & + \frac{1-2^{-(N+1)C_2}}{2(1-2^{-(N+1)C_2}) - (1-2^{-NC_2})}\frac{C_2}{C_1}\left(N - 2^{-NC_2}\frac{1-2^{-NC_2}}{1-2^{-2NC_2}}2N\right)\end{aligned} \tag{3.158}$$

Where $\mathsf{E}[\rho_\theta(0)-1] = K-1$ for uniform prior distribution. □

The part of the proof that adds a factor $r_h = 1-2\epsilon$ for uses the bound on $\tau_0 + S$ is more a sketch than a rigorous mathematical proof. A more rigorous proof of the bound on $\tau_0 + S$ is left as future research.

An approximation, which is neither an upper nor a lower bound, can be obtained by setting N as the ceiling, not unlike an actual system that only stops when the threshold

$\Pr(\hat{\theta} \neq \theta) \leq \epsilon$ is attained. Depending on the crossing value, this larger N might not be needed, which could partially be offset by the other minimization and optimality assumptions on the expression.

Proof of claim 4. Need to prove that $\rho_i(T) = 0.5$ minimizes $f(\rho_n)$, even with the assumption that $p_r^n = 1$. For simplicity, let $x = (1 - rho_n)$, then, $x \in (1-q, \frac{1}{2}]$, and the proof consists of showing that $x = \frac{1}{2}$ minimizes $f(x)$, which is defined by:

$$f(x) = -x\frac{1}{C}\log_2\frac{q}{qx + p(1-x)} + \frac{1-x}{C}\log_2(2(1-x)) + x\mathsf{E}[S_n^{bad}]\,. \tag{3.159}$$

To show that $f(x)$ decreases with x, it suffices to show that $\frac{d}{dx}f(x) > 0$.

$$\begin{aligned}
\frac{d}{dx}f(x) &= -\frac{1}{C}\log_2\frac{q}{qx + p(1-x)} + \frac{C_2}{C_1}\left(1 - 2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right) && (3.160)\\
&\quad - \frac{1}{C}\log_2(2(1-x)) + x\frac{\log_2(e)}{C}\frac{q-p}{qx+p(1-x)} - (1-x)\frac{\log_2(e)}{C}\frac{1}{1-x} && (3.161)\\
&= -\frac{1}{C}\log_2(2q) + \frac{1}{C}\log(2p + 2(q-p)x) - \frac{1}{C}\log_2(2(1-x)) && (3.162)\\
&\quad + \frac{C_2}{C_1}\left(1 - 2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right) && (3.163)\\
&\quad + (q-p)\frac{\log_2(e)}{C}\frac{x}{x(q-p)+p} - \frac{\log_2(e)}{C} && (3.164)\\
&\leq -\frac{1}{C}\log_2(2q) + \frac{1}{1-2p} - \log_2(e)\frac{2p}{C} - \frac{C_2}{C_1}2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N) && (3.165)\\
&= \frac{q\log_2(q) + p\log_2(p) + 1 - \log_2(q) - 1 + 2p\log_2(2q)}{C(1-2p)} - \log_2(e)\frac{2p}{C} && (3.166)\\
&\quad - \frac{C_2}{C_1}2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N) && (3.167)\\
&= \frac{p(-\log_2(q)+\log_2(p)+2\log_2(2q))}{C(1-2p)} - \log_2(e)\frac{2p}{C} - \frac{C_2}{C_1}\frac{2^{-NC_2}-2^{-(N+1)C_2}}{1-2^{-(N+1)C_2}}(1+N)\\
&= \frac{p(2+\log_2(pq))}{C(1-2p)} - \log_2(e)\frac{2p}{C} - \frac{C_2}{C_1}2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N) && (3.168)
\end{aligned}$$

Note that:

$$\log_2(pq) = \log_2(e)\ln(p(1-p)) \tag{3.169}$$

$$\frac{d}{dp}\log_2(pq) = \log_2(e)\frac{1-2p}{p(1-p)} \tag{3.170}$$

Then, it is increasing with $p \in (0, \frac{1}{2})$, with a maximum of -2 at $p = \frac{1}{2}$, and thus, all terms are non-positive. This proves that $x = \frac{1}{2}$ minimizes the correction term. □

3.14 Proof of Lemmas 1-7

Proof of Lemma 1 Equation (3.46). Equation (3.46) of Lemma 1 is the expectation of τ, while (3.45) is the expectation of only the communication phase time $T = \sum_{n=1}^{\infty} T^{(n)}$, that removes from the total time τ the time when the process is in the confirmation phase. The proof equation (3.46) is simpler and helpful to understand the proof of (3.45), and thus is provided first, but will only be used for the proof of the converse bound of Thm. 4. Need to show that:

$$\mathsf{E}[\tau] = \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}^\epsilon} \Pr(Y^t = y^t \mid \theta = i)\,. \tag{3.171}$$

First the expectation $\mathsf{E}[\tau]$ is expressed as $\mathsf{E}_\Omega[\mathsf{E}[\tau \mid \Theta = i]]$:

$$\mathsf{E}[\tau] = \sum_{i=1}^{M} \Pr(\theta = i)\mathsf{E}[\tau \mid \theta = i] \tag{3.172}$$

$$= \sum_{i=1}^{M} \Pr(\theta = i) \sum_{t=1}^{\infty} \Pr(\tau > t \mid \theta = i) \tag{3.173}$$

$$= \sum_{i=1}^{M} \Pr(\theta = i) \sum_{t=1}^{\infty} \mathsf{E}[\mathbb{1}_{(\tau > t)} | \theta = i]\,. \tag{3.174}$$

In (3.173) the tail sum formula for expectations is used, and in (3.174), the probability $\Pr(\tau > t \mid \theta = i)$ is expressed as the expectation of an indicator. Next the definition of expectation is used to expand the $\mathsf{E}[\mathbb{1}_{(\tau>t)}|\theta = i]$:

$$\mathsf{E}[\tau] = \sum_{i=1}^{M} \Pr(\theta{=}i) \sum_{t=1}^{\infty} \sum_{y^t \in \{0,1\}^t} \mathbb{1}_{(y^t \in \mathcal{Y}_{(\tau>t)})} \Pr(Y^t = y^t \mid \theta = i) \tag{3.175}$$

$$= \sum_{i=1}^{M} \Pr(\theta{=}i) \sum_{t=1}^{\infty} \sum_{y^t \in \mathcal{Y}_{(\tau>t)}} \Pr(Y^t = y^t \mid \theta = i) \tag{3.176}$$

$$= \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \cup_{t=0}^{\infty} \mathcal{Y}_{(\tau>t)}} \Pr(Y^t = y^t \mid \theta = i) \tag{3.177}$$

$$= \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}^\epsilon} \Pr(Y^t = y^t \mid \theta = i)\,. \tag{3.178}$$

The indicator in (3.175) is one if $y^t \in \mathcal{Y}_{(\tau>t)}$ and zero if $y^t \in \{0,1\}^t \setminus \mathcal{Y}_{(\tau>t)}$. Equation (3.176) only includes set $\mathcal{Y}_{(\tau>t)}$ where the indicator is a constant one, and is omitted. In (3.177) the sums over t and over $\mathcal{Y}_{(\tau>t)}$ are expressed as a single sum over the union $\cup_{t=0}^{\infty}\mathcal{Y}_{(\tau>t)}$ of all $\mathcal{Y}_{(\tau>t)}$, which is the definition of $\mathcal{Y}^\epsilon$. Replacing the union by the definition $\mathcal{Y}^\epsilon$ completes the proof. □

Proof of Lemma 1 Equation (3.45). Need to show that:

$$\mathsf{E}[T] = \sum_{i=1}^{M} \Pr(\theta = i) \sum_{y^t \in \mathcal{Y}_i^\epsilon} \Pr(Y^t = y^t \mid \theta = i) = \sum_{i=1}^{M} \sum_{y^t \in \mathcal{Y}_i^\epsilon} \Pr(Y^t = y^t, \theta = i)\,. \tag{3.179}$$

The proof of Lemma 1 makes use of four special sets that need to be introduced first. For each sequence $y^t \in \mathcal{Y}^\epsilon$, define a set $N_i(y^t)$ as the set of time values $t_0^{(1)}, t_0^{(2)}, \ldots, t_0^{(n)} \leq t$, where message i begins an interval with $U_i(t) < 0$. This includes time zero and all the times s where from time $s-1$ to s, message i transitions from $U_i(s-1) \geq 0$ to $U_i(s) < 0$, i.e. the

decoder falls back from confirmation phase to communication phase.

$$N_i(y^t) \triangleq \{0\} \cup \{s \leq t : U_i(s) < 0, U_i(s-1) \geq 0\}\,. \tag{3.180}$$

Define the set $\mathcal{Y}^{\epsilon}_{(i,n)}$ of sequences y^t for which the following are all true: 1) the decoder has not stopped, 2) the decoder has entered the confirmation phase for message i n times, and 3) the decoder is not in the confirmation phase for message i at time t, where the sequence ends.

$$\mathcal{Y}^{\epsilon}_{(i,n)} \triangleq \{y^t \in \mathcal{Y}^{\epsilon}_i : \left|N_i(y^t)\right| = n, U_i(t) < 0\}\,. \tag{3.181}$$

For each sequence $y^t \in \mathcal{Y}^{\epsilon}_{(i,n)}$, define the set $\mathcal{Y}^{\epsilon}_{(i,n)}(y^s)$, as the subset of sequences from $\mathcal{Y}^{\epsilon}_{(i,n)}$ that have the sequence y^s as a prefix.

$$\mathcal{Y}^{\epsilon}_{(i,n)}(y^s) \triangleq \{y^t \in \mathcal{Y}^{\epsilon}_{(i,n)} \mid t \geq s, y^t_{1:s} = y^s\} \tag{3.182}$$

Finally, let $\mathcal{B}^{\epsilon}_{(i,n)}$ be the set containing only the sequences where the final received symbol y_t is the symbol for which the decoder resumes the communication phase for message i for the n^{th} time, or the empty string, that is:

$$\mathcal{B}^{\epsilon}_{(i,n)} \triangleq \{y^t \in \mathcal{Y}^{\epsilon}_{(i,n)} \big|\ t \in N_i(y^t)\}\,. \tag{3.183}$$

Each $y^t \in \mathcal{B}^{\epsilon}_{(i,n)}$, sets an initial condition for the communication phase where $U_i(t) < 0$, so that $T^{(n)} \geq 1$, that is $t = t_0^{(n)}$, as defined in (3.20). By the property of conditional expectation, $\mathsf{E}[T]$ is given by:

$$\mathsf{E}[T] = \sum_{i=1}^{M} \Pr(\theta = i)\mathsf{E}[T \mid \theta = i]\,. \tag{3.184}$$

Next, the resulting expression is explicitly written as a function of all the possible initial

conditions, for each of the communication phase rounds n. That is, the set $\mathcal{B}^{\epsilon}_{(i,n)}$:

$$\sum_{i=1}^{M}\Pr(\theta=i)\mathsf{E}[T\,|\,\theta=i]=\sum_{i=1}^{M}\Pr(\theta=i)\mathsf{E}\left[\left(\sum_{n=1}^{\infty}T^{(n)}\right)\Big|\theta=i\right] \tag{3.185}$$

$$=\sum_{i=1}^{M}\Pr(\theta=i)\sum_{n=1}^{\infty}\mathsf{E}\left[T^{(n)}\Big|\theta=i\right] \tag{3.186}$$

$$=\sum_{i=1}^{M}\Pr(\theta=i)\sum_{n=1}^{\infty}\sum_{y^s\in\mathcal{B}^{\epsilon}_{(i,n)}}\Pr(Y^s=y^s\,|\,\theta=i)\mathsf{E}\left[T^{(n)}\Big|Y^s=y^s,\theta=i\right]. \tag{3.187}$$

In step (3.188), the expectation $\mathsf{E}\left[T^{(n)}\Big|Y^s=y^s,\theta=i\right]$ in (3.187) is transformed using the tail sum formula for expectations, and then, in (3.189), as an expectation of the indicator of $\{T^{(n)}>r\}$. Since $T^{(n)}$ is a random function of $Y^t=Y^sY^r$, where $Y^r\in\{0,1\}^r$, given by $\mathbb{1}_{T^{(n)}>r}=\mathbb{1}_{Y^sY^r\in\mathcal{Y}^{\epsilon}_{(i,n)}}$, equation (3.190) follows:

$$\mathsf{E}\left[T^{(n)}\Big|Y^s=y^s,\theta=i\right]=\sum_{r=0}^{\infty}\Pr(T^{(n)}>r|Y^s=y^s,\theta=i) \tag{3.188}$$

$$=\sum_{r=0}^{\infty}\mathsf{E}[\mathbb{1}_{T^{(n)}>r}|Y^s=y^s,\theta=i] \tag{3.189}$$

$$=\sum_{r=0}^{\infty}\mathsf{E}[\mathbb{1}_{Y^{s+r}\in\mathcal{Y}^{\epsilon}_{(i,n)}(y^s)}|Y^s=y^s,\theta=i]\,. \tag{3.190}$$

Expanding the expectation in (3.190), equation (3.191) is obtained. Since the indicator in (3.191) is 0 outside $\mathcal{Y}^{\epsilon}_{(i,n)}$ and 1 inside, it is omitted in (3.192), where only values of y^sz^r that intersect with $\mathcal{Y}^{\epsilon}_{(i,n)}$ are considered. Since $\cup_{r=1}^{\infty}\{\{0,1\}^r\cap\mathcal{Y}^{\epsilon}_{(i,n)}(y^s)\}=\mathcal{Y}^{\epsilon}_{(i,n)}(y^s)$, then (3.192)

follows.

$$\sum_{r=0}^{\infty} \mathsf{E}[\mathbb{1}_{[Y^{s+r}\in\mathcal{Y}^{\epsilon}_{(i,n)}(y^s)]} | Y^s = y^s, \theta = i]$$

$$= \sum_{r=0}^{\infty} \sum_{z^r\in\{0,1\}^r} \mathbb{1}_{Y^{s+r}\in\mathcal{Y}^{\epsilon}_{(i,n)}(y^s)} \Pr(Y^{s+r}=y^s z^r \mid Y^s=y^s, \theta=i) \tag{3.191}$$

$$= \sum_{y^s z^r \in \cup_{r=1}^{\infty}\{\{0,1\}^r \cap \mathcal{Y}^{\epsilon}_{(i,n)}(y^s)\}} \Pr(Y^{s+r}=y^s z^r \mid Y^s=y^s, \theta=i) \tag{3.192}$$

$$= \sum_{y^{s+r}\in\mathcal{Y}^{\epsilon}_{(i,n)}(y^s)} \Pr(Y^{s+r} = y^s z^r \mid Y^s = y^s, \theta = i)\,. \tag{3.193}$$

The product of conditional probabilities $\Pr(Y^s=y^s \mid \theta=i) \cdot \Pr(Y^{s+r} = y^s z^r \mid Y^s = y^s, \theta = i)$ from (3.187) and (3.193) is given by $\Pr(Y^{s+r} = y^s z^r \mid \theta = i)$. Replacing the expectation in (3.187) by (3.193) the inner-most sum in (3.187) becomes (3.194). The summation in (3.194) is over $\mathcal{Y}^{\epsilon}_{(i,n)}(y^s)$ for each y^s in $\mathcal{B}^{\epsilon}_{(i,n)}$, and every sequence in $\mathcal{Y}^{\epsilon}_{(i,n)}$ has a prefix in $\mathcal{B}^{\epsilon}_{(i,n)}$, that is: $\bigcup_{y^s\in\mathcal{B}^{\epsilon}_{(i,n)}} \mathcal{Y}^{\epsilon}_{(i,n)}(y^s) = \mathcal{Y}^{\epsilon}_{(i,n)}$.

$$\sum_{y^s\in\mathcal{B}^{\epsilon}_{(i,n)}} \Pr(Y^s=y^s \,|\, \theta=i)\mathsf{E}\left[T^{(n)}\middle| Y^s=y^s, \theta=i\right] = \sum_{y^s\in\mathcal{B}^{\epsilon}_{(i,n)}} \sum_{y^{s+r}\in\mathcal{Y}^{\epsilon}_{(i,n)}(y^s)} \Pr(Y^{s+r}=y^{s+r} \,|\, Y^s=y^s, \theta=i) \tag{3.194}$$

$$= \sum_{t=1}^{\infty} \sum_{y^t\in\mathcal{Y}^{\epsilon}_{(i,n)}} \Pr(Y^t = y^t \mid \theta = i)\,. \tag{3.195}$$

The expectation $\mathsf{E}\left[T^{(n)}\middle|\theta = i\right]$ in equation (3.186) is now replaced by (3.195), to write (3.185) as (3.196). In (3.197) the two sums are consolidated into a single sum of all y^t in the union

$\cup_{n=0}^{\infty}\mathcal{Y}_{(i,n)}^{\epsilon}$, of the sets $\mathcal{Y}_{(i,n)}^{\epsilon}$, over all n:

$$\sum_{i=1}^{M}\Pr(\theta{=}i)\sum_{n=1}^{\infty}\sum_{r=0}^{\infty}\mathsf{E}[\mathbb{1}_{T^{(n)}>r}|\theta=i]=\sum_{i=1}^{M}\Pr(\theta{=}i)\sum_{n=1}^{\infty}\sum_{y^t\in\mathcal{Y}_{(i,n)}^{\epsilon}}\Pr(Y^t=y^t\mid\theta=i) \tag{3.196}$$

$$=\sum_{i=1}^{M}\Pr(\theta=i)\sum_{y^t\in\cup_{n=0}^{\infty}\mathcal{Y}_{(i,n)}^{\epsilon}}\Pr(Y^t=y^t\mid\theta=i)\,. \tag{3.197}$$

To conclude the proof, note that the union $\cup_{n=0}^{\infty}\mathcal{Y}_{(i,n)}^{\epsilon}$ is the set $\mathcal{Y}_i^{\epsilon}$ defined in the statement of the Lemma 1. □

Proof of Lemma 2, inequality 3.50. Need to show that the following inequality holds:

$$C\sum_{i=1}^{M}\Pr(\theta{=}i)\sum_{y^t\in\mathcal{Y}_i^{\epsilon}}\Pr(Y^t=y^t\mid\theta{=}i)\leq\sum_{i=1}^{M}\Pr(\theta{=}i)\sum_{y^t\in\mathcal{Y}_i^{\epsilon}}V_i(y^t)\Pr(Y^t=y^t\mid\theta{=}i) \tag{3.198}$$

Let the set $\mathcal{Y}_i^{\epsilon}$ be partitioned into $\mathcal{A}_\epsilon$ and $\mathcal{Y}_i^{\epsilon}\setminus\mathcal{A}_\epsilon$. Then, the sum in the right side of (3.198), which is the right side of inequality (3.50) of Lemma 2, can be split into a sum over $\mathcal{A}_\epsilon$, right side of (3.199), and a sum over the sets $\mathcal{Y}_i^{\epsilon}\setminus\mathcal{A}_\epsilon$, expression (3.200), as follows:

$$\sum_{i=1}^{M}\Pr(\theta{=}i)\sum_{y^t\in\mathcal{Y}_i^{\epsilon}}\Pr(Y^t{=}y^t\mid\theta{=}i)V_i(y^t)=\sum_{i=1}^{M}\Pr(\theta{=}i)\sum_{y^t\in\mathcal{A}_\epsilon}\Pr(Y^t{=}y^t\mid\theta{=}i)V_i(y^t) \tag{3.199}$$

$$+\sum_{i=1}^{M}\Pr(\theta{=}i)\sum_{y^t\in\mathcal{Y}_i^{\epsilon}\setminus\mathcal{A}_\epsilon}\Pr(Y^t{=}y^t\mid\theta{=}i)V_i(y^t)\,. \tag{3.200}$$

For $y^t\in\mathcal{Y}_i^{\epsilon}\setminus\mathcal{A}_\epsilon:\ \exists j\neq i$ s.t. $U_j(t)\geq 0$ and $U_i(t)<0$. By *Claim* (2), the set S_0 needs to be the singleton $S_0=\{j\}$, in order to satisfy constraint (3.25). Then, the partitions S_0 and S_1 also satisfy Naghshvar's SED constraint [NJW15], which guarantees that $\mathsf{E}[U_i(t+1)-U_i(t)\mid Y^t=y^t,\theta=i]\geq C$ for every $i\in\Omega$, by inequality (3.8). To prove the Lemma, it suffices to show that the bound holds also for (3.199). The product of conditional probabilities: $\Pr(\theta=i)$ and $\Pr(Y^t=y^t\mid\theta=i)$ in (3.199) is equal to $\Pr(Y^t=y^t,\theta=i)$, and can be

factored into $\Pr(Y^t = y^t)\Pr(\theta = i \mid Y^t = y^t)$. Since $0 < V_i(y^t) \leq C_2$, and $\mathcal{A}_\epsilon$ does not depend on i, the summation order in (3.199) can be reversed, to obtain:

$$\sum_{i=1}^{M}\sum_{y^t\in\mathcal{A}_\epsilon}\Pr(Y^t{=}y^t)\Pr(\theta{=}i\,|Y^t{=}y^t)V_i(y^t){=}\sum_{y^t\in\mathcal{A}_\epsilon}\Pr(Y^t{=}y^t)\sum_{i=1}^{M}\Pr(\theta{=}i\,|Y^t{=}y^t)V_i(y^t)\,. \tag{3.201}$$

The inner sum in (3.201) is lower bounded by constraint (3.26) of Thm. 1. Note that the probability $\Pr(\theta = i \mid Y^t = y^t)$ in (3.201) is just $\rho_i(y^t)$. The term $V_i(y^t)$ is replaced by its definition (3.47), to obtain:

$$\sum_{i=1}^{M}\Pr(\theta = i \mid Y^t = y^t)V_i(y^t) = \sum_{i\in\Omega}\rho_i(y^t)E[U_i(t{+}1){-}U_i(t)|Y^t{=}y^t,\theta{=}i] \tag{3.202}$$

$$= E[U_\theta(t+1) - U_\theta(t)|Y^t = y^t] \geq C\,. \tag{3.203}$$

Note that the right of (3.202) is the definition of $E[U_\theta(t{+}1){-}U_\theta(t)|Y^t = y^t]$. Equation (3.204) follows by applying bound C from constraint (3.26) in Thm. 1 to (3.203). The marginal probability $\Pr(Y^t = y^t)$ is replaced by the sum over the joint probability $\Pr(Y^t = y^t, \theta = i)$ in (3.205), and (3.206) follows from the chain rule of probabilities:

$$\sum_{y^t\in\mathcal{A}_\epsilon}\Pr(Y^t = y^t)\sum_{i=1}^{M}\rho_i(y^t)V_i(y^t) \geq \sum_{y^t\in\mathcal{A}_\epsilon}\Pr(Y^t = y^t)C \tag{3.204}$$

$$= C\sum_{i=1}^{M}\sum_{y^t\in\mathcal{A}_\epsilon}\Pr(Y^t = y^t, \theta = i) \tag{3.205}$$

$$= C\sum_{i=1}^{M}\Pr(\theta = i)\sum_{y^t\in\mathcal{A}_\epsilon}\Pr(Y^t = y^t \mid \theta = i)\,. \tag{3.206}$$

The proof is complete. □

Proof of Lemma 2 inequality (3.51).

$$\text{Define:} \qquad t_0 \triangleq \sum_{i=1}^{M} \Pr(\theta{=}i) \sum_{y^t \in \mathcal{A}_\epsilon} \Pr(Y^t = y^t \mid \theta{=}i) \tag{3.207}$$

$$\text{Need to show that:} \qquad t_0 \geq \frac{1}{C} \sum_{i=1}^{M} \Pr(\theta{=}i) \sum_{y^t \in \mathcal{A}_\epsilon} \Psi_i(y^t) \Pr(Y^t{=}y^t \mid \theta{=}i) \tag{3.208}$$

$$\text{Definition (3.48):} \qquad \Psi_i(y^t) \triangleq \mathsf{E}[\log_2(\rho_i(y^{t+1})) - \log_2(\rho_i(y^t)) | Y^t = y^t, \theta = i] \,. \tag{3.209}$$

Since both $\Pr(\theta{=}i)$ and $\Pr(Y^t = y^t \mid \theta{=}i)$ are probabilities, the order of summation in the definition of t_0 can be reversed:

$$t_0 = \sum_{i=1}^{M} \Pr(\theta{=}i) \sum_{y^t \in \mathcal{A}_\epsilon} \Pr(Y^t = y^t \mid \theta{=}i) \tag{3.210}$$

$$= \sum_{i=1}^{M} \sum_{y^t \in \mathcal{A}_\epsilon} \Pr(Y^t = y^t, \theta{=}i) \tag{3.211}$$

$$= \sum_{y^t \in \mathcal{A}_\epsilon} \sum_{i=1}^{M} \Pr(Y^t = y^t, \theta{=}i) \tag{3.212}$$

$$= \sum_{y^t \in \mathcal{A}_\epsilon} \Pr(Y^t = y^t) \sum_{i=1}^{M} \Pr(\theta{=}i \mid Y^t = y^t) \tag{3.213}$$

$$= \sum_{y^t \in \mathcal{A}_\epsilon} \Pr(Y^t = y^t) \sum_{i=1}^{M} \rho_i(y^t) \,. \tag{3.214}$$

Note that $\sum_{i=1}^{M} \rho_i(y^t) = 1$. The next part of the proof consists of showing that:

$$C \geq \mathsf{E}[\Psi_\theta(y^t) \mid Y^t = y^t] \tag{3.215}$$

$$= \sum_{i=1}^{M} \Pr(\theta = i \mid Y^t = y^t) \mathsf{E}[\rho_i(y^{t+1})) - \log_2(\rho_i(y^t)) \mid Y^t = y^t, \theta = i] \tag{3.216}$$

By the definition of $\mathcal{A}_\epsilon$, $y^t \in \mathcal{A}_\epsilon \implies \rho_i(y^t) < \frac{1}{2}, \forall i \in \Omega$. To show that equation (3.216) holds, it suffices to show that it is a concave function, with a unique maximum of C, when

$\Delta = 0$. First, note that,given y^t, the value of $\rho_i(y^t)$ is constant, an can thus be extracted from the expectation. The ratio $\frac{\rho_i(y^{t+1})}{\rho_i(y^t)}$ is exactly the weight update coefficient $W_i(y_{t+1}) \triangleq \frac{\Pr(Y_{t+1}=y_{t+1}|\theta=i,Y^t=y^t)}{\Pr(Y_{t+1}=y_{t+1}|Y^t=y^t)}$. Since $\{Y^t = y^t\}$ fully determines the partitions S_0 and S_1, the top probability $\Pr(Y_{t+1} = y_{t+1} \mid \theta = i, Y^t = y^t)$ is q if $i \in S_{y_{t+1}}$ and p otherwise. The bottom probability $\Pr(Y_{t+1} = y_{t+1} \mid Y^t = y^t)$ is $qP_{y_{t+1}} + p(1 - P_{y_{t+1}})$. When the events $\{i \in S_0\}$ and $\{i \in S_1\}$ are considered separately, it can be expressed by:

$$\sum_{i=1}^{M} \rho_i(y^t)\mathsf{E}[\rho_i(y^{t+1})) - \log_2(\rho_i(y^t)) \mid Y^t = y^t, \theta = i] \tag{3.217}$$

$$= \sum_{i=\in S_0} \rho_i(y^t) \left(q \log_2 \frac{q}{P_0 q + P_1 p} + p \log_2 \frac{p}{P_0 p + P_1 q} \right) \tag{3.218}$$

$$+ \sum_{i=\in S_1} \rho_i(y^t) \left(q \log_2 \frac{q}{P_1 q + P_0 p} + p \log_2 \frac{p}{P_1 p + P_0 q} \right) \tag{3.219}$$

$$\sum_{i=1}^{M} \rho_i(y^t)\mathsf{E}[\rho_i(y^{t+1})) - \log_2(\rho_i(y^t)) \mid Y^t = y^t, \theta = i] \tag{3.220}$$

$$= \sum_{i=\in S_0} \rho_i(y^t) \left(q \log_2 \frac{2q}{1 + \Delta(q - p)} + p \log_2 \frac{2p}{1 - \Delta(q - p)} \right) \tag{3.221}$$

$$+ \sum_{i=\in S_1} \rho_i(y^t) \left(q \log_2 \frac{2q}{1 - \Delta(q - p)} + p \log_2 \frac{2p}{1 + \Delta(q - p)} \right) \tag{3.222}$$

$$= C - \frac{1 + \Delta}{2} \left(q \log_2(1 + \Delta(q - p)) + p \log_2(1 - \Delta(q - p)) \right) \tag{3.223}$$

$$- \frac{1 - \Delta}{2} \left(q \log_2(1 - \Delta(q - p)) + p \log_2(1 + \Delta(q - p)) \right) \tag{3.224}$$

$$= C + f(\Delta, p, q) \tag{3.225}$$

Where $f(\Delta, p, q)$ is given by:

$$f(\Delta, p, q) = -\frac{1+\Delta}{2}\left(q\log_2(1+\Delta(q-p)) + p\log_2(1-\Delta(q-p))\right) \tag{3.226}$$

$$-\frac{1-\Delta}{2}\left(q\log_2(1-\Delta(q-p)) + p\log_2(1+\Delta(q-p))\right) \tag{3.227}$$

$$= -(q^{\frac{1+\Delta}{2}} + p^{\frac{1-\Delta}{2}})\log_2(1+\Delta(q-p)) - (q^{\frac{1-\Delta}{2}} + p^{\frac{1+\Delta}{2}})\log_2(1-\Delta(q-p)) \tag{3.228}$$

$$= -\frac{1+(q-p)\Delta}{2}\log_2(1+\Delta(q-p)) - \frac{1-(q-p)\Delta}{2}\log_2(1-\Delta(q-p)) \tag{3.229}$$

$$= -\frac{1+(q-p)\Delta}{2\ln(2)}\ln(1+\Delta(q-p)) - \frac{1-(q-p)\Delta}{2\ln(2)}\ln(1-\Delta(q-p)) \tag{3.230}$$

To find the maximum of $\mathsf{E}[\Psi_\theta(y^t) \mid Y^t = y^t]$, it suffices to maximize $f(\Delta, p, q)$, or $g(\Delta, p, q) \triangleq 2\ln(2)f(\Delta)$, the latter which ignores the additive constant C of the expectation, as well as the multiplicative constant $\frac{1}{\ln(2)}$ in $f(\Delta, p, q)$. The proof consists of showing that $g(\Delta, p, q)$ is concave, with a maximum of 0 at $\Delta = 0$. It suffices to show that the first derivative of $g(\Delta, p, q)$ is zero at $\Delta = 0$ and that the second derivative is negative.

$$g(\Delta, p, q) = -(1+(q-p)\Delta)\ln(1+\Delta(q-p)) - (1-(q-p)\Delta)\ln(1-\Delta(q-p)) \tag{3.231}$$

$$\frac{d}{d\Delta}g(\Delta, p, q) = -(q-p)\ln(1+\Delta(q-p)) + (q-p)\ln(1-\Delta(q-p)) \tag{3.232}$$

$$-(1+(q-p)\Delta)\frac{(q-p)}{1+\Delta(q-p)} - (1-(q-p)\Delta)\frac{-(q-p)}{1-\Delta(q-p)} \tag{3.233}$$

$$= -(q-p)\ln(1+\Delta(q-p)) + (q-p)\ln(1-\Delta(q-p)) \tag{3.234}$$

$$-(q-p)\frac{1+(q-p)\Delta}{1+\Delta(q-p)} + \frac{1-(q-p)\Delta}{1-\Delta(q-p)} \tag{3.235}$$

$$= -(q-p)\ln(1+\Delta(q-p)) + (q-p)\ln(1-\Delta(q-p)) \tag{3.236}$$

The first derivative of $g(\Delta, p, q)$ is zero at $\Delta = 0$, since $\ln(1) = 0$. Next the second derivative

is derived:

$$\frac{d^2}{d\Delta^2}g(\Delta,p,q) = \frac{d}{d\Delta}(-\ln(1+\Delta(q-p)) + \ln(1-\Delta(q-p))) \tag{3.237}$$

$$= -\frac{q-p}{1+\Delta(q-p)} - \frac{q-p}{1-\Delta(q-p)} \tag{3.238}$$

$$= -(q-p)\frac{1-\Delta(q-p)+(1+\Delta(q-p))}{1-\Delta^2(q-p)^2} \tag{3.239}$$

$$= -2\frac{(q-p)}{1-\Delta^2(q-p)^2} \tag{3.240}$$

For every $\Delta(q-p) \in (0,1)$ the second derivative is negative, which proves the concavity. Thus, the expectation $\mathsf{E}[\Psi_\theta(y^t) \mid Y^t = y^t]$ has a maximum of C at $\Delta = 0$. To complete the proof, not that for each $i \in \Omega$:

$$i \in S_0 \implies \Psi_i(y^t) = C - q\log_2(1+\Delta(q-p)) - p\log_2(1-\Delta(q-p)) \tag{3.241}$$

$$i \in S_1 \implies \Psi_i(y^t) = C - q\log_2(1-\Delta(q-p)) - p\log_2(1+\Delta(q-p)) \tag{3.242}$$

In both cases $\Psi_i(y^t)$ has a minimum of 0 at $\Delta = 1$ and $\Delta = -1$ respectively. Thus, $\Psi_i(y^t)$ is non-negative, which allows to revert the order of summation. Furthermore, these extremes are only achieved with either $S_0 = \emptyset$ or $S_1 = \emptyset$, which is not a valid partitioning, and thus $\Psi_i(y^t) > 0$. The proof is complete. □

Proof of Lemma 3, equation (3.52). Define $V_i(y^t) \triangleq \mathsf{E}[U_i(t+1) - U_i(t) \mid Y^t = y^t, \theta = i]$ to rewrite the left side of (3.52) as (3.243):

$$\sum_{y^t \in \mathcal{Y}_i^\epsilon} \mathsf{E}[U_i(t+1) - U_i(t) \,|\, Y^t = y^t, \theta = i] \Pr(Y^t = y^t \,|\, \theta = i) = \sum_{y^t \in \mathcal{Y}_i^\epsilon} V_i(y^t) \Pr(Y^t = y^t \,|\, \theta = i)\,. \tag{3.243}$$

An equivalent form of $\mathcal{Y}_i^\epsilon$, given by $\cup_{n=0}^{\infty} \mathcal{Y}_{(i,n)}^\epsilon$, is used to write (3.243) as the left side of (3.245). This form was also used in the proof of Lemma 1. The sum over the union is again

split into a double sum, see right side of (3.245), first over n, and then over $\mathcal{Y}^{\epsilon}_{(i,n)}$:

$$\sum_{y^t\in\mathcal{Y}^{\epsilon}_i} V_i(y^t)\Pr(Y^t{=}y^t\,|\,\theta=i) = \sum_{y^t\in\cup_{n=0}^{\infty}\mathcal{Y}^{\epsilon}_{(i,n)}} V_i(y^t)\,\Pr(Y^t{=}y^t\,|\,\theta=i) \tag{3.244}$$

$$= \sum_{n=1}^{\infty}\sum_{y^t\in\mathcal{Y}^{\epsilon}_{(i,n)}} V_i\,(y^t)\,\Pr(Y^t{=}y^t\,|\,\theta{=}i)\,. \tag{3.245}$$

The set $\mathcal{Y}^{\epsilon}_{(i,n)}$ is a subset of $\cup_{t=0}^{\infty}\{0,1\}^t$, therefore, it can be expressed a union of all the intersections over n: $\mathcal{Y}^{\epsilon}_{(i,n)} = \cup_{t=0}^{\infty}\{\mathcal{Y}^{\epsilon}_{(i,n)}\cap\{0,1\}^t\}$. This new form is used to rewrite the sum in (3.245), see in left of (3.246). In the right of (3.246), the intersections with $\mathcal{Y}^{\epsilon}_{(i,n)}$ are removed, and replaced by the indicator of $\mathcal{Y}^{\epsilon}_{(i,n)}$:

$$\sum_{t=0}^{\infty}\sum_{y^t\in\mathcal{Y}^{\epsilon}_{(i,n)}\cap\{0,1\}^t} V_i(y^t)\,\Pr(Y^t{=}y^t\mid\theta{=}i) = \sum_{t=0}^{\infty}\sum_{y^t\in\{0,1\}^t}\mathbb{1}_{y^t\in\mathcal{Y}^{\epsilon}_{(i,n)}} V_i(y^t)\,\Pr(Y^t{=}y^t\mid\theta{=}i)\,. \tag{3.246}$$

Recall that $V_i(y^t) = \mathsf{E}\,[U_i(t+1){-}U_i(t)\,|\,Y^t=y^t,\theta{=}i]$ from (3.47). Also recall from (7.2) that $U_i(t) = U_i(Y^t)$, a random function of Y^t. Define $D_i(Y^{t+1}) \triangleq U_i(Y^{t+1}) - U_i(Y^t)$, to expand $V_i(y^t)$ as:

$$\mathsf{E}[U_i(t{+}1){-}U_i(t)\mid Y^t{=}y^t,\theta{=}i] = \sum_{z\in\{0,1\}} D_i(y^tz)\,\Pr(Y_{t+1}{=}z\mid Y^t{=}y^t,\theta{=}i)\,. \tag{3.247}$$

The product of the probabilities in (3.246) and (3.247) is given by $\Pr(Y^{t+1}=y^tz\mid\theta=i)$. Then, replace $V_i(y^t)$ in (3.246) with (3.247), to obtain the left side of (3.248). The equality in (3.248) follows by the definition of expectation:

$$\sum_{t=0}^{\infty}\sum_{y^{t+1}\in\{0,1\}^{t+1}} D_i(y^{t+1})\mathbb{1}_{y^t\in\mathcal{Y}^{\epsilon}_{(i,n)}}\Pr(Y^{t+1}{=}y^{t+1}\,|\,\theta{=}i) = \sum_{t=0}^{\infty}\mathsf{E}[D_i(Y^{t+1})\mathbb{1}_{Y^t\in\mathcal{Y}^{\epsilon}_{(i,n)}}\mid\theta{=}i]\,. \tag{3.248}$$

Then, expand $D_i(Y^t)$ using its definition, to rewrite (3.248) in (3.249), and use linearity of expectations in (3.250). The indicator $\mathbb{1}_{Y^t\in\mathcal{Y}^{\epsilon}_{(i,n)}}$ is zero before time $t=t_0^{(n)}$, is zero after time

$t = t_0^{(n)} + T^{(n)} - 1$, and is one when t is between the two times: $t_0^{(n)} \leq t < t = t_0^{(n)} + T^{(n)}$. Thus, the limits of summation are adjusted in (3.251), and the indicator of $\mathcal{Y}_{(i,n)}^{\epsilon}$ removed. Note that the times $t_0^{(n)}$ and $T^{(n)}$ are themselves random variables. Lastly, observe that (3.251) is a telescopic sum, which is replaced by its end points in (3.252):

$$\sum_{t=0}^{\infty} \mathsf{E}[D_i(Y^{t+1}) \mathbb{1}_{Y^t \in \mathcal{Y}_{(i,n)}^{\epsilon}} \mid \theta = i] = \sum_{t=0}^{\infty} \mathsf{E}\left[\left(U_i\left(Y^{t+1}\right) - U_i\left(Y^t\right) \right) \mathbb{1}_{Y^t \in \mathcal{Y}_{(i,n)}^{\epsilon}} \middle| \theta = i \right] \tag{3.249}$$

$$= \mathsf{E}\left[\sum_{t=0}^{\infty} \left(U_i\left(Y^{t+1}\right) - U_i\left(Y^t\right) \right) \mathbb{1}_{Y^t \in \mathcal{Y}_{(i,n)}^{\epsilon}} \middle| \theta = i \right] \tag{3.250}$$

$$= \mathsf{E}\left[\sum_{t=t_0^{(n)}}^{T^{(n)}+t_0^{(n)}-1} (U_i(Y^{t+1}) - U_i(Y^t)) \middle| \theta = i \right] \tag{3.251}$$

$$= \mathsf{E}\left[U_i\left(t_0^{(n)} + T^{(n)}\right) - U_i\left(t_0^{(n)}\right) | \theta = i \right] . \tag{3.252}$$

Finally, the inner most summation in (3.245) is replaced with (3.252):

$$\sum_{n=1}^{\infty} \sum_{y^t \in \mathcal{Y}_{(i,n)}^{\epsilon}} V_i\left(y^t\right) \Pr(Y^t = y^t \mid \theta = i) = \sum_{n=1}^{\infty} \mathsf{E}\left[U_i\left(t_0^{(n)} + T^{(n)}\right) - U_i\left(t_0^{(n)}\right) | \theta = i \right] . \tag{3.253}$$

The proof is complete. □

Proof of lemma 3, equation (3.53). Equation (3.53) differs from (3.52) in that it the logs of the posteriors $\log_2(\rho_i(y^t))$, $i \in \Omega$ replace the log likelihood ratio $U_i(t)$, $i \in \Omega$. Need to show that:

$$\sum_{y^t \in \mathcal{Y}_i} \mathsf{E}[\log_2\left(\frac{\rho_i(y^{t+1})}{\rho_i(y^t)}\right) | Y^t = y^t, \theta = i] \Pr(Y^t = y^t | \theta = i) = \mathsf{E}[\log_2\left(\frac{\rho_i(y^{T_0})}{\rho_i(y^0)}\right) \mid \theta = i] . \tag{3.254}$$

The same proof used in (3.52) can be used, but only the first time $T_1 = T^{(1)}$ is needed, since only the first crossing of the correct message into the confirmation phase is considered, and the difference $D_i(Y^{t+1})$ needs to be replaced. The new definition is given by: $D_i(Y^{t+1}) \triangleq$

$\rho_i(Y^{t+1}) - \rho_i(Y^t)$. $\square$

Proof of Lemma 4, fall back probability p_f of equation (3.54). The confirmation phase starts at a time t of the form T_n defined in (3.19), at which the transmitted message i attains $U_i(T_n) \geq 0$ and $U_i(T_n - 1) < 0$. Then, similar to the product martingale in [Dur19a], the process $\zeta_i(t)$, $t \geq T_n$, is a martingale respect to $\mathcal{F}_t = \sigma(Y^t)$, where:

$$\zeta_i(t) = \left(\frac{p}{q}\right)^{\frac{U_i(t)}{C_2}} . \tag{3.255}$$

Note that $U_i(t)$ is a biased random walk, see the Markov Chain in [YPA21], with $U_i(t) = U_i(T_n) + \sum_{m=T_n}^{t} \xi_m$, where ξ_m is an R.V. distributed according to:

$$\xi_m = \begin{cases} +C_2 & \text{w.p. } q \\ -C_2 & \text{w.p. } p \end{cases} , \tag{3.256}$$

To prove that $\zeta_i(t)$ is a martingale, need to show that $\mathsf{E}[\zeta_i(t+1) \mid \mathcal{F}_t] = \zeta_i(t)$:

$$\mathsf{E}[\zeta_i(t+1) \mid \mathcal{F}_t] = \zeta_i(t) \left(p \left(\frac{p}{q}\right)^{-1} + q \left(\frac{p}{q}\right)^{1} \right) = \zeta_i(t) \left(p + q\right) = \zeta_i(t) . \tag{3.257}$$

Let S_n be the first time t at which decoding terminates, if $U_i(t) = U_i(T_n) + NC_2$, or a fall back occurs, if $U_i(t) = U_i(T_n) - C_2 < 0$, that is:

$$S_n \triangleq \min\{t \geq T_n : U_i(t) \in \{U_i(T_n) - C_2, U_i(T_n) + NC_2\}\} . \tag{3.258}$$

Then, the process $\zeta_i(t \wedge S_n)$ is a two-side bounded martingale and:

$$\mathsf{E}[\zeta_i(S_n)] = p_f \left(\frac{p}{q}\right)^{\frac{U_i(T_n)}{C_2}-1} + (1-p_f)\left(\frac{p}{q}\right)^{\frac{U_i(T_n)}{C_2}+N} \tag{3.259}$$

$$\mathsf{E}[\zeta_i(T_n)] = \left(\frac{p}{q}\right)^{\frac{U_i(T_n)}{C_2}} \tag{3.260}$$

By Doob's optional stopping theorem [Dur19b], $\mathsf{E}[\zeta_i(S_n)]$ is equal to $\mathsf{E}[\zeta_i(T_n)]$. Let the fall back probability be $p_f \triangleq \Pr(U_i(S_n) = U_i(T_n) - C_2 \mid t = T_n)$, then, the value of p_f is obtained from equations (3.259) and (3.260), by setting both right sides equal. The term $(p/q)^{U_i(T_n)/C_2}$ is cancelled from both sides in (3.261), the terms with p_f are collected in (3.262), and the value of p_f is obtained in (3.263).

$$1 = p_f \frac{q}{p} + (1-p_f)\left(\frac{p}{q}\right)^N \tag{3.261}$$

$$0 = p_f \frac{q}{p}\left(1 - \left(\frac{p}{q}\right)^{N+1}\right) - \left(1 - \left(\frac{p}{q}\right)^N\right) \tag{3.262}$$

$$p_f = \frac{p}{q}\frac{1-\left(\frac{p}{q}\right)^N}{1-\left(\frac{p}{q}\right)^{N+1}}\,. \tag{3.263}$$

Since p_f is just a function of N and p, then it is the same constant, as long as $\{\theta = i\}$ independent of the value of $i \in \Omega$, and independent of the index $n \in \mathbb{N}$. The definition of C_2 in equation (7.5), given by $C_2 = \log_2\left(\frac{q}{p}\right)$ is used to replace $\frac{p}{q}$ and obtain p_f in terms of C_2:

$$p_f = 2^{-\log_2(\frac{q}{p})}\frac{1-2^{-N\log_2(\frac{q}{p})}}{1-2^{-(N+1)\log_2(\frac{q}{p})}} = 2^{-C_2}\frac{1-2^{-NC_2}}{1-2^{-(N+1)C_2}} \tag{3.264}$$

The proof is complete. □

Proof of Lemma 4 recovery probability p_r of equation (3.56). When the message i in confirmation comprises the entire set S_0, by claim 2 and and constraint (3.24), the magnitude of

the step size taken by $U_i(t)$ every time step is C_2. However, when the wrong message is in confirmation, the encoder sends $X_t = 1$. Thus, $U_i(t)$ moves backwards when $Y_t = X_t$, with probability q, and forward when $Y_t \neq X_t$, with probability p. To obtain the recovery probability, a martingale $\zeta_i(t)$, $t \geq T_n$ is constructed, starting at time $t = T_n$. The martingale $\zeta_i(t)$ is similar that used in the proof of (3.54), but $\frac{q}{p}$ replaces $\frac{p}{q}$, and is given by:

$$\zeta_i(t) = \left(\frac{q}{p}\right)^{\frac{U_i(t)}{C_2}} \tag{3.265}$$

The process $U_i(t)$ is again a biased random walk: $U_i(t) = U_i(T_n) + \sum_{m=T_n}^{t} \xi_m$, where $U_i(T_n)$ is the value of $U_i(t)$ when crossing into the confirmation phase, and ξ_m is an R.V. distributed according to:

$$\xi_m = \begin{cases} -C_2 & \text{w.p. } q \\ +C_2 & \text{w.p. } p \end{cases} \tag{3.266}$$

To show that $\zeta_i(t)$ is a martingale, it suffices to show that $\mathsf{E}[\zeta_i(t+1) \mid \mathcal{F}_t] = \zeta_i(t)$:

$$\mathsf{E}[\zeta_i(t+1) \mid \mathcal{F}_t] = \zeta_i(t) \left(p \left(\frac{q}{p}\right)^{1} + q \left(\frac{q}{p}\right)^{-1} \right) = \zeta_i(t) \left(p + q\right) = \zeta_i(t) \tag{3.267}$$

Let S_n be again a stopping time as defined in (3.258). Then, the martingale $\zeta_i(t \wedge S_n)$ is two-side bounded. Define the recovery probability by $p_r \triangleq \Pr(U_i(S_n) = U_i(T_n) - C_2)$, the

probability that the confirmation phase ends in a recovery, then:

$$\mathsf{E}[\zeta_i(S_n)] = \mathsf{E}[\zeta_i(T_n)] = \left(\frac{q}{p}\right)^{\frac{U_i(T_n)}{C_2}} \tag{3.268}$$

$$= p_r \left(\frac{q}{p}\right)^{\frac{U_i(T_n)}{C_2}-1} + (1-p_r)\left(\frac{q}{p}\right)^{\frac{U_i(T_n)}{C_2}+N} \tag{3.269}$$

$$\left(\frac{q}{p}\right)^{\frac{U_i(T_n)}{C_2}} = p_r \left(\frac{q}{p}\right)^{\frac{U_i(T_n)}{C_2}-1} + (1-p_r)\left(\frac{q}{p}\right)^{\frac{U_i(T_n)}{C_2}+N} \tag{3.270}$$

$$1 = p_r \frac{p}{q} + \left(\frac{q}{p}\right)^{N} - p_r \left(\frac{q}{p}\right)^{N} \tag{3.271}$$

$$0 = p_r \frac{p}{q}\left(1 - \left(\frac{q}{p}\right)^{N+1}\right) - \left(1 - \left(\frac{q}{p}\right)^{N}\right) \tag{3.272}$$

Solving for p_r:

$$p_r = \frac{q}{p}\frac{1-\left(\frac{q}{p}\right)^{N}}{1-\left(\frac{q}{p}\right)^{N+1}} = 2^{C_2}\frac{1-2^{NC_2}}{1-2^{(N+1)C_2}} \tag{3.273}$$

$$= \frac{1-2^{-NC_2}}{1-2^{-(N+1)C_2}} \tag{3.274}$$

The probability that the process ends in a wrong decoding is given by $1-p_r$, where:

$$(1-p_r) = 1 - \frac{1-2^{-NC_2}}{1-2^{-(N+1)C_2}} = \frac{1-2^{-(N+1)C_2}-1+2^{-NC_2}}{1-2^{-(N+1)C_2}} \tag{3.275}$$

$$= \frac{2^{-NC_2}-2^{-(N+1)C_2}}{1-2^{-(N+1)C_2}} = 2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}} \tag{3.276}$$

The proof is complete. □

Proof of Lemma 4 Corollary 1. Suppose now, that the correct message is in the confirmation phase, and replace N by $m+n$. Suppose that the process starts at a time t_n with a value $U_i(t_n) = U_i(t_n) + nC_2$, and let the stopping time S_n be the time at which $U_i(t) \in \{U_i(t_n) - nC_2, U_i(t_n) + (n+m)C_2\}$. Define p_m as the probability that the process ends at

$U_i(t) = U_i(t_n) + (n+m)C_2$, which could be associated with a correct decoding, and p_n as the probability that the process ends at $U_i(t) = U_i(t_n) - nC_2$, a fall back. Then, the martingale $\zeta(t)$ in (3.255), used in the proof of the fall back probability p_f (3.54), can be used to compute p_m, and p_{n^-}:

$$\mathsf{E}[\zeta_i(S_n)] = \mathsf{E}[\zeta_i(t_n)] = \left(\frac{p}{q}\right)^{\frac{U_i(t_n)}{C_2}} \tag{3.277}$$

$$= p_m \left(\frac{p}{q}\right)^{\frac{U_i(t_n)}{C_2}+m} + (1-p_m)\left(\frac{p}{q}\right)^{\frac{U_i(t_n)}{C_2}-n} \tag{3.278}$$

$$\left(\frac{p}{q}\right)^{\frac{U_i(t_n)}{C_2}} = p_m \left(\frac{p}{q}\right)^{\frac{U_i(t_n)}{C_2}+m} + (1-p_m)\left(\frac{p}{q}\right)^{\frac{U_i(T_n)}{C_2}-n} \tag{3.279}$$

$$\left(\frac{p}{q}\right)^{n} = p_m \left(\frac{p}{q}\right)^{n+m} - p_m + 1 \tag{3.280}$$

$$0 = p_m \left(\left(\frac{p}{q}\right)^{n+m} - 1\right) - \left(\left(\frac{q}{p}\right)^{n} - 1\right) \tag{3.281}$$

$$p_m = \frac{\left(\frac{p}{q}\right)^{n} - 1}{\left(\frac{p}{q}\right)^{n+m} - 1} \tag{3.282}$$

$$= \frac{1 - 2^{-nC_2}}{1 - 2^{(n+m)C_2}} \tag{3.283}$$

And $p_{n^-} = 1 - p_m$, given by:

$$p_{n^-} = 1 - \frac{1 - 2^{-nC_2}}{1 - 2^{-(n+m)C_2}} = \frac{2^{-(n+m)C_2} - 2^{-nC_2}}{1 - 2^{-(n+m)C_2}} \tag{3.284}$$

$$= 2^{-nC_2} \frac{1 - 2^{-mC_2}}{1 - 2^{-(n+m)C_2}} \tag{3.285}$$

The proof is complete. □

Proof of Lemma 5, equation (3.63). Let $\beta_n(t) = U_i(t) - (t - t_0^{(n)} - T^{(n)})C_1$ for $t > T_n$, where

$T_n = t_0^{(n)} + T^{(n)}$, and let S_n be a stopping time, defined by:

$$S_n \triangleq \min\{t \geq T_n : \beta_n(t) \in \{\beta_n(T_n) - C_2, \beta_n(T_n) + NC_2\}\} - T_n \,. \tag{3.286}$$

Then, $\beta_n(t)$ is a martingale, and $\beta_n(t \wedge S_n)$ is a two-side bounded martingale, and at $t = S_n$, $\beta(S_n) \in \{U_i(0) - C_2, U_i(0) + NC_2\}$. The expected duration $\mathsf{E}[S_n] - T_n$, of the n-th round of the correct confirmation phase, can be computed from $\mathsf{E}[S_n]$, using the martingale $\beta_n(t)$ as follows:

$$\begin{aligned}
\mathsf{E}[\beta(S_n)] = \beta_n(T_n) = U_i(T_n) &= \mathsf{E}[U_i(S_n)] - C_1\mathsf{E}[S_n] && (3.287)\\
&= p_f(U_i(T_n) - C_2) + (1 - p_f)(U_i(T_n) + NC_2) - \mathsf{E}[S_n]C_1 && (3.288)\\
U_i(T_n) &= U_i(T_n) - p_f C_2 + (1 - p_f)NC_2 - \mathsf{E}[S_n]C_1 && (3.289)\\
C_1\mathsf{E}[S_n] &= -C_2 p_f + (1 - p_f)NC_2 && (3.290)\\
\mathsf{E}[S_n] &= \frac{C_2}{C_1}\left((1 - p_f)N - p_f\right) && (3.291)
\end{aligned}$$

Then $\mathsf{E}[S_n] - T_n = \mathsf{E}[S_m] - T_n \quad \forall n, m \geq 1$, i.e. it is a constant $\mathsf{E}[S] \triangle \mathsf{E}[S_n] - T_n$, that does not depend on n. The confirmation phase will happen by average of $\sum_{j=0}^{\infty} p_f^i = \frac{1}{1-p_f}$ times. Let $\tau - T$ denote the total spent by the correct message in the confirmation phase, then:

$$\mathsf{E}[\tau - T] = \mathsf{E}[S]\frac{1}{1 - p_f} = (N - \frac{p_f}{1 - p_f})\frac{C_2}{C_1} = \left(N - 2^{-C_2}\frac{1 - 2^{-NC_2}}{1 - 2^{-C_2}}\right)\frac{C_2}{C_1} \tag{3.292}$$

The proof is complete. □

Proof of Lemma 5, expected wrong confirmation time, equation (3.65). Similar to the proof of equation (3.63), Let $\beta_n(t) = U_i(t) + (t - t_0^{(n)} - T^{(n)})C_1$ for $t > T_n = t_0^{(n)} + T^{(n)}$, and let S_n be the stopping time defined in (3.286). Then, $\beta_n(t)$ is a martingale, and $\beta_n(t \wedge S_n)$ is a two-side bounded martingale, and at $t = S_n$, $\beta(S_n) \in \{U_i(0) - C_2, U_i(0) + NC_2\}$. The expected duration of the n-th round of a wrong confirmation phase is $\mathsf{E}[S_n] - T_n$, where $\mathsf{E}[S_n]$

is given by:

$$\mathsf{E}[\beta(S_n)] = \beta_n(T_n) = U_i(T_n) = \mathsf{E}[U_i(S_n)] + C_1\mathsf{E}[S_n] \tag{3.293}$$

$$= p_r(U_i(T_n) - C_2) + (1-p_r)(U_i(T_n) + NC_2) + \mathsf{E}[S_n]C_1 \tag{3.294}$$

$$U_i(T_n) = U_i(T_n) - p_rC_2 + (1-p_r)NC_2 + \mathsf{E}[S_n]C_1 \tag{3.295}$$

$$C_1\mathsf{E}[S_n] = C_2p_r - (1-p_r)NC_2 \tag{3.296}$$

$$\mathsf{E}[S_n] = \frac{C_2}{C_1}\left(p_r - (1-p_r)N\right) \tag{3.297}$$

$$= \frac{C_2}{C_1}\left(p_r(1+N) - N\right) \tag{3.298}$$

$$= \frac{C_2}{C_1}\left(\frac{1-2^{-NC_2}}{1-2^{-(N+1)C_2}}(1+N) - N\right) \tag{3.299}$$

$$= \frac{C_2}{C_1}\left(\frac{1-2^{-(N+1)C_2}-2^{-NC_2}+2^{-(N+1)C_2}}{1-2^{-(N+1)C_2}}(1+N) - N\right) \tag{3.300}$$

$$= \frac{C_2}{C_1}\left(N+1-N-2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right) \tag{3.301}$$

$$= \frac{C_2}{C_1}\left(1-2^{-NC_2}\frac{1-2^{-C_2}}{1-2^{-(N+1)C_2}}(1+N)\right) \tag{3.302}$$

And is a constant that does not depend on the round n and times T_n or $t_0^{(n)}$.

□

Proof of Lemma 5 Corollary 2. Suppose that at some time $t = t' > T_n$, $U_i(t') = U_I(T_N) + nC_2$, and define the martingale $\beta_n(t) = U_i(t) - (t-t')C_1$ for $t > t'$, similar to the martingale in the proof of (3.63). Let the stopping time S_{n+m} by:

$$S_{n+m} \triangleq \min\{t \geq t' : \beta_n(t) \in \{\beta_n(T_n), \beta_n(T_n) + (n+m)C_2\}\} - t'\,, \tag{3.303}$$

and let p_m and p_{n^-} be as defined in Lemma 4 Corollary 1:

$$p_m = \frac{1 - 2^{-nC_2}}{1 - 2^{-(n+m)C_2}} \tag{3.304}$$

$$p_{n^-} = 2^{-nC_2} \frac{1 - 2^{-mC_2}}{1 - 2^{-(n+m)C_2}} \tag{3.305}$$

Then, $\beta(t)$, $t \geq t'$ is a two side bounded martingale, and:

$$\mathsf{E}[\beta(S_{n+m})] = \beta_n(t') = U_i(T_n) + nC_2 = \mathsf{E}[U_i(S_{n+m})] - C_1\mathsf{E}[S_{n+m}] \tag{3.306}$$

$$U_i(T_n) + nC_2 = (1 - p_m)U_i(T_n) + p_m(U_i(T_n) + (n+m)C_2) - \mathsf{E}[S_{n+m}]C_1 \tag{3.307}$$

$$nC_2 = p_m(n+m)C_2 - \mathsf{E}[S_{n+m}]C_1 \tag{3.308}$$

$$C_1\mathsf{E}[S_{n+m}] = p_m(n+m)C_2 - nC_2 \tag{3.309}$$

$$\mathsf{E}[S_{n+m}] = \frac{C_2}{C_1}\left(mp_m - n(1-p_m)\right) = \frac{C_2}{C_1}\left(m(1-p_{n^-}) - np_{n^-}\right) \tag{3.310}$$

$$= \frac{C_2}{C_1}\left(p_m(m+n) - n\right) = \frac{C_2}{C_1}\left(m - p_{n^-}(m+n)\right) \tag{3.311}$$

$$= \frac{C_2}{C_1}\left(\frac{1 - 2^{-nC_2}}{1 - 2^{-(n+m)C_2}}(m+n) - n\right) \tag{3.312}$$

$$= \frac{C_2}{C_1}\left(m - 2^{-nC_2}\frac{1 - 2^{-mC_2}}{1 - 2^{-(n+m)C_2}}(m+n)\right). \tag{3.313}$$

And if $m = n = N$, then, using p_N and p_{N^-}, then $\mathsf{E}[S_{2N}]$ is given by:

$$\mathsf{E}[S_{2N}] = \frac{C_2}{C_1}\left(\frac{1 - 2^{-NC_2}}{1 - 2^{-2NC_2}}2N - N\right) \tag{3.314}$$

$$= \frac{C_2}{C_1}\left(N - 2^{-NC_2}\frac{1 - 2^{-NC_2}}{1 - 2^{-2NC_2}}2N\right) \tag{3.315}$$

The proof is complete. □

Proof of Lemma 6. Start by conditioning the expectation in the left side of equation (3.68), in Lemma 6, on the events $\{T^{(n)} > 0, \theta = i\}$, $\{T^{(n)} = 0, \theta = i\}$, and $\{T^{(n)} < 0, \theta = i\}$, whose union results in the original conditioning event, $\{\theta = i\}$, to express the original conditional

probability using Bayes rule:

$$\mathsf{E}[U_i(T_n)-U_i(t_0^{(n)}) \mid \theta = i] = \mathsf{E}[U_i(T_n)-U_i(t_0^{(n)}) \,|\, T^{(n)}>0, \theta=i] \Pr(T^{(n)}>0 \,|\, \theta=i) \tag{3.316}$$

$$+ \mathsf{E}[U_i(T_n)-U_i(t_0^{(n)}) \,|\, T^{(n)}=0, \theta=i] \Pr(T^{(n)}=0 \,|\, \theta=i) \tag{3.317}$$

$$+ \mathsf{E}[U_i(T_n)-U_i(t_0^{(n)}) \,|\, T^{(n)}<0, \theta=i] \Pr(T^{(n)}<0 \,|\, \theta=i)\,. \tag{3.318}$$

Note that $T^{(n)}$ is non-negative, and thus $\Pr(T^{(n)} < 0) = 0$, and term (3.318) vanishes. When $T^{(n)} = 0$, then $T_n = t_0^{(n)} + T^{(n)} = t_0^{(n)}$ so that $U_i(T_n) = U_i(t_0^{(n)})$, and the term (3.317) also vanishes, leaving only the first term conditioned on $\{T^{(n)} > 0, \theta = i\}$. Let $\mathcal{C}(t_0^{(n)})$ be the event that message i enters confirmation after time $t_0^{(n)}$, rather than another message $j \neq i$ ending the process by attaining $U_j(t) \geq \log_2(1-\epsilon)-\log_2(\epsilon)$, that is: $\mathcal{C}(t_0^{(n)}) \triangleq \{\exists t > t_0^{(n)} : U_i(t) \geq 0\}$. Then, the probability $\Pr(T^{(n+1)} \geq 0 \mid \theta = i)$ can be expressed as:

$$\Pr(T^{(n+1)}\geq 0 \,|\, \theta=i) = \Pr(T^{(n+1)} \geq 0 \,|\, T^{(n)}>0, \mathcal{C}(t_0^{(n)}), \theta=i) \Pr(T^{(n)}>0, \mathcal{C}(t_0^{(n)}) \,|\, \theta=i). \tag{3.319}$$

Note that the first probability in the right side of (3.319) is just the fall back probability p_f computed in Lemma 4. The last probability in (3.319) can be also expressed as a product of conditional probabilities, see (3.320). In (3.320) note that event $\mathcal{C}(t_0^{(n)})$ is the event that an n-th confirmation phase phase occurs, which implies that a preceding n-th communication phase round occurs. Then, $\mathcal{C}(t_0^{(n)}) \implies T^{(n)} > 0$, and the first factor in the product of probabilities in (3.320) vanishes:

$$\Pr(T^{(n)}>0, \mathcal{C}(t_0^{(n)}) \,|\, \theta=i) = \Pr(T^{(n)}>0 \,|\, \mathcal{C}(t_0^{(n)}), \theta=i) \Pr(\mathcal{C}(t_0^{(n)}) \,|\, \theta=i) \tag{3.320}$$

$$= \Pr(\mathcal{C}(t_0^{(n)}) \,|\, \theta=i)\,. \tag{3.321}$$

Combine (3.319) and (3.320) to obtain:

$$\Pr(T^{(n+1)}>0 \mid \theta=i) = \Pr(\mathcal{C}(t_0^{(n)}) \mid \theta=i) p_f\,, \tag{3.322}$$

and bound $\Pr(\mathcal{C}(t_0^{(n)}) \mid \theta = i)$ as follows:

$$\Pr(\mathcal{C}(t_0^{(n)}) \,|\, \theta = i) = \Pr(\mathcal{C}(t_0^{(n)}) \,|\, T^{(n)} > 0, \theta = i) \Pr(T^{(n)} > 0 \,|\, \theta = i) \leq \Pr(T^{(n)} > 0 \,|\, \theta = i)\,. \tag{3.323}$$

Then, recursively bound $\Pr(T^{(n+1)} > 0 \mid \theta = i)$ by $\Pr(T^{(n)} > 0 \mid \theta = i)p_f$ using (3.322) and (3.323). For $n \geq 1$, this results in the general bound:

$$\Pr(T^{(n)} \geq 0 \mid \theta = i) \leq p_f^{n-1}\,. \tag{3.324}$$

Using (3.324), the expectation $\mathsf{E}[U_i(T_n) - U_i(t_0^{(n)}) \mid \theta = i]$, in (3.316), is bounded by:

$$\mathsf{E}[U_i(T_n) - U_i(t_0^{(n)}) \mid \theta = i] \leq \mathsf{E}[U_i(T_n) - U_i(t_0^{(n)}) \,|\, T^{(n)} > 0, \theta = i] p_f^{n-1}\,. \tag{3.325}$$

The value of $U_i(t_0^{(n)})$ at $n = 1$, when $t_0^{(1)} = 0$ is a constant that can be directly computed for every i from the initial distribution. Using (3.325), the second sum in the left side of equation (3.68) can be bounded by:

$$\sum_{n=1}^{\infty} \mathsf{E}[U_i(T_n) - U_i(t_0^{(n)}) \,|\, \theta = i] \tag{3.326}$$

$$\leq \mathsf{E}[U_i(T^{(1)}) - U_i(0) \,|\, T^{(1)} > 0, \theta = i] p_f^0 + \sum_{n=2}^{\infty} \mathsf{E}[U_i(T_n) - U_i(t_0^{(n)}) \,|\, T^{(n)} > 0, \theta = i] p_f^{n-1} \tag{3.327}$$

$$= -U_i(0) + \sum_{n=1}^{\infty} \mathsf{E}[U_i(T_n) \,|\, T^{(n)} > 0, \theta = i] p_f^{n-1} - \sum_{n=2}^{\infty} \mathsf{E}[U_i(t_0^{(n)}) \mid T^{(n)} > 0, \theta = i] p_f^{n-1}\,. \tag{3.328}$$

The conditioning event $\{T^{(n)} > 0\}$ implies events $\{T^{(m)} > 0\}$ for $m = 1, \ldots, n$ because if $T^{(m)} = 0$ means the process has stopped and no further communication rounds occur. Event $\{T^{(n)} > 0\}$ also implies that the n-th round of communication occurs, and therefore $U_i(t_0^{(n)})$

is given by the previous crossing value $U_i(T_{n-1})$ minus C_2 by constraint (3.25), then:

$$\sum_{n=2}^{\infty}\mathsf{E}[U_i(t_0^{(n)}) \mid T^{(n)}>0,\theta=i]p_f^{n-1}=\sum_{n=2}^{\infty}\mathsf{E}[U_i(T_{n-1})-C_2 \mid T^{(n)}>0,\theta=i]p_f^{n-1} \tag{3.329}$$

$$=\sum_{n=1}^{\infty}\mathsf{E}[U_i(T_n)-C_2 \mid T^{(n+1)}>0,\theta=i]p_f^{n}\geq\sum_{n=1}^{\infty}\mathsf{E}[U_i(T_n)-C_2 \mid T^{(n)}>0,\theta=i]p_f^{n}\,. \tag{3.330}$$

The following inequality was used in (3.330), and the proof is provided in Sec. 3.22:

$$\mathsf{E}[U_i(T_n) \mid T^{(n+1)}>0,\theta=i]\geq\mathsf{E}[U_i(T_n) \mid T^{(n)}>0,\theta=i]\,, \tag{3.331}$$

From (3.330) it follows that:

$$-\sum_{n=1}^{\infty}\mathsf{E}[U_i(T_n)-C_2 \mid T^{(n+1)}>0,\theta=i]p_f^{n}\leq-\sum_{n=1}^{\infty}\mathsf{E}[U_i(T_n)-C_2 \mid T^{(n)}>0,\theta=i]p_f^{n} \tag{3.332}$$

$$=-\sum_{n=1}^{\infty}\mathsf{E}[p_f(U_i(T_n)-C_2) \mid T^{(n)}>0,\theta=i]p_f^{n-1}. \tag{3.333}$$

Note that in (3.333), a factor p_f from the power p_f^n is moved inside the expectation. Replace (3.328) by (3.333) to upper bound (3.326) by:

$$\sum_{n=1}^{\infty}\mathsf{E}[U_i(T_n)-U_i(t_0^{(n)}) \mid \theta=i]\leq-U_i(0) \tag{3.334}$$

$$+\sum_{n=1}^{\infty}\mathsf{E}[U_i(T_n) \mid T^{(n)}>0,\theta=i]p_f^{n-1}-\sum_{n=1}^{\infty}\mathsf{E}[p_f(U_i(T_n)-C_2) \mid T^{(n)}>0,\theta=i]p_f^{n-1} \tag{3.335}$$

$$=\sum_{n=1}^{\infty}\mathsf{E}[U_i(T_n)-p_f(U_i(T_n)-C_2) \mid T^{(n)}>0,\theta=i]p_f^{n-1}-U_i(0)\,. \tag{3.336}$$

The expectation in (3.336) combines the two sums in (3.335) by subtracting $p_f(U_i(T_n)-C_2)$ from $U_i(T_n)$. The first term $U_i(T_n)$ is the value of $U_i(t)$ at the communication-phase stopping time $t=T_n$. In the second term $p_f(U_i(T_n)-C_2)$, the difference $U_i(T_n)-C_2$ is the unique value that $U_i(t_0^{(n+1)})$ can take once the n-th confirmation phase round starts at a point $U_i(T_n)$.

Equation (3.336) is an important intermediate result in the proof of Thm. 2. This is because when considering the process $U_i'(t)$, the starting value of each communication-phase round $U_i'(t_0^{(n+1)})$ is still that of the original process $U_i(T_n) - C_2$, and therefore, the argument of the expectation would change to $U_i'(T_n') - p_f(U_i(T_n) - C_2)$. To proof Lemma 6, it suffices to bound (3.336), for which the sum in (3.336) is written as:

$$\sum_{n=1}^{\infty} \mathsf{E}[U_i(T_n)(1-p_f)+p_f C_2)\,|T^{(n)}>0, \theta=i]p_f^{n-1} \\ = \sum_{n=1}^{\infty} \mathsf{E}[U_i(T_n) \mid T^{(n)}>0, \theta=i](1-p_f)p_f^{n-1} + \sum_{n=1}^{\infty} C_2 p_f^n\,. \quad (3.337)$$

By constraint (3.23), $U_i(T_n) \leq U_i(T_n - 1) + C_2$, and since $U_i(T_n - 1) < 0$, then, $U_i(T_n)$ is bounded by C_2. Thus, the expectation $\mathsf{E}[U_i(T_n) \mid T^{(n)}>0, \theta=i]$ is also bounded by C_2. Then (3.337) is bounded by:

$$\sum_{n=1}^{\infty} C_2(1-p_f)p_f^{n-1} + \sum_{n=1}^{\infty} C_2 p_f^n = C_2 + \frac{p_f}{1-p_f} C_2 \quad (3.338)$$

Finally, the left side of equation (3.68) in Lemma 6, (the left side of (3.339)), is upper bounded using the bounds (3.336) and (3.338) on the inner sum (3.332) as follows:

$$\sum_{i=1}^{M} \Pr(\theta=i) \sum_{n=1}^{\infty} \mathsf{E}[U_i(T_n) - U_i(t_0^{(n)}) \mid \theta=i] \leq \sum_{i=1}^{M} \Pr(\theta=i) \left(\frac{p_f}{1-p_f} C_2 + C_2 - U_i(0) \right) \quad (3.339)$$

$$= 2^{-C_2} \frac{1-2^{-NC_2}}{1-2^{-C_2}} C_2 + C_2 - \mathsf{E}[U_i(0)]\,. \quad (3.340)$$

To transition from (3.339) to (3.340), the definition of p_f from Lemma 4 is used. The proof of Lemma 6 is complete. □

Proof of lemma7. The wrong communication phase starts at some time $t = t'$, when a message $i \in \Omega \setminus \theta$ has attained a posterior $U_i(t') \geq 0$. Let $r_1, r_2, \ldots, r_n$ denote the expected times that the wrong message i spends in each state k, that is: $U_i(t) = U_i(t') + kC_2$. Let

the last state in the Markov Chain be n, from which there is no return. Note that, $U_i(t)$ takes a step back with probability q and a step forward with probability p, Therefore, each time i visits a state k, it spends one time unit in state k, and then moves to state $k-1$ with probability q and to state $k+1$ with probability p. The time r_0 spent at state 0 is given by $1+qr_1$, and since there is no return from state n, the time r_n spent in state n is just $r_{n-1}p$. The times spent on all intermediate states: $r_2, r_3, \ldots, r_{n-1}$ can be expressed recursively by $r_k = pr_{k-1} + qr_k$, starting by state r_1 as follows:

$$r_1 = pr_0 + qr_2 \tag{3.341}$$

$$r_2 = pr_1 + qr_3 \tag{3.342}$$

$$r_3 = pr_2 + qr_4 \tag{3.343}$$

$$r_4 = pr_3 + qr_5 \tag{3.344}$$

$$\ldots = \ldots \tag{3.345}$$

$$r_k = pr_{k-1} + qr_{k+1} \tag{3.346}$$

$$\ldots = \ldots \tag{3.347}$$

$$r_{n-2} = pr_{n-3} + qr_{n-1} \tag{3.348}$$

$$r_{n-1} = pr_{n-2} + qr_n \tag{3.349}$$

$$r_n = pr_{n-1} \tag{3.350}$$

Adding all the left sides and right sides separately the following inequality is obtained:

$$\sum_{i=1}^{n} r_i = \sum_{i=0}^{n-1} pr_i + \sum_{i=2}^{n} qr_i \tag{3.351}$$

$$= pr_0 + pr_1 + \sum_{i=2}^{n-1} r_i + qr_n \tag{3.352}$$

Then subtract the two sums:

$$\sum_{i=1}^{n} r_i - \sum_{i=2}^{n-1} = r_0 + pr_1 + qr_n \tag{3.353}$$

$$r_1 + r_n = pr_0 + pr_1 + qr_n \tag{3.354}$$

$$r_1(1-p) + pr_n = pr_0 \tag{3.355}$$

$$r_1 = r_0\frac{p}{1-p} - pr_n \tag{3.356}$$

$$r_1 = r_0\frac{p}{q} - pr_n \tag{3.357}$$

The process ends if the state n is reached. In such case, a single time unit is spent on state n. Then, r_n is also the probability of reaching state n: $r_n = \Pr(\exists t > t' : U_i(t) = U_i(t) + nC_2)$. This is the probability $(1 - p_r)$ of no recovery in Lemma 4 equation 3.57, with $N = n$:

$$r_n = 2^{-nC_2}\frac{1 - 2^{-C_2}}{1 - 2^{-(n+1)C_2}} \tag{3.358}$$

When the number of states goes to infinity, the probability r_n goes to zero:

$$\lim_{n\to\infty} r_n = \lim_{n\to\infty} 2^{-nC_2}\frac{1 - 2^{-C_2}}{1 - 2^{-(n+1)C_2}} \tag{3.359}$$

$$= 0 \tag{3.360}$$

$$\lim_{n\to\infty} r_1 = \frac{p}{q} \tag{3.361}$$

Solving for the other states:

$$r_1 = r_0\frac{p}{q} = r_0 p + qr_2 \qquad \implies r_2 = r_0(\frac{p}{q} - p)\frac{1}{q} = r_0\frac{p - pp}{q^2} = r_0\frac{p(1-q)}{q^2} = r_0\frac{p^2}{q^2} \tag{3.362}$$

Assume $r_k = r_0 \frac{p^k}{q^k}$ and inductively compute r_{k+1}:

$$r_k = r_0 \frac{p^k}{q^k} = r_0 p r_{k-1} + q r_{k+1} = r_0 \frac{p^k}{q^{k-1}} + q r_{k+1} = r_0 \frac{p^k q}{q^k} + q r_{k+1} \tag{3.363}$$

$$q r_{k+1} = r_0 \frac{p^k}{q^k} - r_0 \frac{p^k q}{q^k} \tag{3.364}$$

$$r_{k+1} = r_0 \frac{1}{q} \frac{p^k (1-q)}{p^k} = r_0 \frac{p^{k+1}}{q^{k+1}} \tag{3.365}$$

It only remains to determine r_0. For an infinite Markov chain with states $0, 1, 2, \ldots$, the message returns to s_0 after leaving to state s_1 *a.s.*. Then the time at s_0 is given by:

$$r_0 = \sum_{i=0}^{\infty} p^i = \frac{1}{1-p} = \frac{1}{q}\,. \tag{3.366}$$

The proof is complete. □

Proof of Lemma 8. Suppose that the process in in the wrong confirmation phase. That is, for $i \in \Omega \setminus \theta$, $U_i(t) \geq 0$. Let t' denote the start of such confirmation phase, then, for the duration of the confirmation phase, $U_i(t) = U_i(t') + nC_2$, forming a Markov Chain. Denote by ρ_n the posterior of the wrong message i, when i is at state n, where $\rho_0 = \rho_i(y^{t'})$. Denote by $W_q(n)$ the log of the weight update of messages in $\Omega \setminus \{i\}$, when $Y_t = X_t = 1$, and by $W_p(n)$ the log of the weight update, when $Y_t = 0 \neq X_t$. The first part of the proof is to show that that $W_q(n+1) = -W_p(n)$, and thus, the wrong confirmation phase becomes a telescopic sum where every update $W_q(n)$ cancels, except for the very first step $W_q(0)$. The update $W_q(0)$ is only used when a recovery occurs, with probability q every time i is at state 0. For the infinite Markov Chain of Lemma 7 this recovery happens a.s. Note that $\{i\}$ is the set S_0 and $S_1 = \Omega \setminus \{i\}$. Thus, $P_0 = \rho_n$ and $P_1 = (1 - \rho_n)$ when i is at state n. The weight updates $W_q(n)$ and $W_p(n)$, of items in S_1 are obtained from the coefficient of $\rho_i(y^t)$

in (2.6) where $S_0 = \{i\}$, and are given by:

$$W_q(n) = \log_2 \frac{q}{q(1-\rho_n) + p\rho_n} \tag{3.367}$$

$$W_p(n) = \log_2 \frac{p}{p(1-\rho_n) + q\rho_n} \tag{3.368}$$

To show that $W_q(n+1) = -W_p(n)$, need to obtain ρ_{n+1} in terms of ρ_n, or equivalently, both in terms of ρ_0:

$$U_i(t') = \log_2 \tfrac{\rho_0}{1-\rho_0} \tag{3.369}$$

$$U_i(t') + nC_2 = \log_2 \tfrac{\rho_n}{1-\rho_n} = \tag{3.370}$$

$$\frac{\rho_n}{1-\rho_n} = \frac{\rho_0}{1-\rho_0}\frac{q^n}{p^n} \tag{3.371}$$

$$\rho_n(1 + \tfrac{\rho_0}{1-\rho_0}\tfrac{q^n}{p^n}) = \frac{\rho_0}{1-\rho_0}\frac{q^n}{p^n} \tag{3.372}$$

$$\rho_n = \frac{\rho_0}{1-\rho_0}\frac{q^n}{p^n}\frac{1}{1 + \frac{\rho_0}{1-\rho_0}\frac{q^n}{p^n}} \tag{3.373}$$

$$= \frac{\rho_0 q^n}{p^n(1-\rho_0) + \rho_n q^n} \tag{3.374}$$

$$1 - \rho_n = \frac{(1-\rho_0)p^n}{p^n(1-\rho_0) + \rho_n q^n} \tag{3.375}$$

Now compute $W_q(n+1)$ and $W_p(n)$:

$$W_p(n) \quad = \log_2 \frac{p}{p(1-\rho_n) + q\rho_n} = \log_2 \frac{p}{p\frac{(1-\rho_0)p^n}{p^n(1-\rho_0)+\rho_0 q^n} + q\frac{\rho_0 q^n}{p^n(1-\rho_0)+\rho_0 q^n}} \tag{3.376}$$

$$= \log_2 \frac{pp^n(1-\rho_0) + \rho_0 pq^n}{p(1-\rho_0)p^n + q\rho_0 q^n} \tag{3.377}$$

$$W_q(n+1) = \log_2 \frac{qp^{n+1}(1-\rho_0) + \rho_0 qq^{n+1}}{q(1-\rho_0)p^{n+1} + p\rho_0 q^{n+1}} = \log_2 \frac{qp^{n+1}(1-\rho_0) + \rho_0 qq^{n+1}}{qp(1-\rho_0)p^n + pq\rho_0 q^n} \tag{3.378}$$

$$= \log_2 \frac{p^{n+1}(1-\rho_0) + \rho_0 q^{n+1}}{(1-\rho_0)pp^n + \rho_0 pq^n} \tag{3.379}$$

$$= -W_p(n)\,. \tag{3.380}$$

At each state, the log of the posterior of the correct message, and every other message in $S_1 = \Omega \setminus \{i\}$, gets the additive update $W_q(n)$ with probability q and $W_p(n)$ with probability p. Then, for each ρ_0 crossing the process leaves the confirmation phase with S_m given by:

$$S = \sum_{m=1}^{\infty} P_m(1-\rho_m)S_m \tag{3.381}$$

$$S_m = \frac{(1-\rho_m)}{q}\sum_{i=0}^{\infty}\frac{p^i}{q^i}(qW_q(n)+pW_p(n)) \tag{3.382}$$

$$S = \sum_{m=1}^{\infty} P_m\frac{(1-\rho_m)}{q}\sum_{i=0}^{\infty}\frac{p^i}{q^i}(qW_q(n)+pW_p(n)) \tag{3.383}$$

Where S_m is given by:

$$S_m = \frac{(1-\rho_m)}{q}\sum_{i=0}^{\infty}\frac{p^i}{q^i}(qW_q(n)+pW_p(n)) \tag{3.384}$$

$$= \frac{(1-\rho_m)}{q}q\sum_{i=0}^{\infty}\frac{p^i}{q^i}W_q(n) + \frac{(1-\rho_m)}{q}p\sum_{i=0}^{\infty}\frac{p^i}{q^i}W_p(n) \tag{3.385}$$

$$= (1-\rho_m)\sum_{i=0}^{\infty}\frac{p^i}{q^i}W_q(n) + (1-\rho_m)\sum_{i=0}^{\infty}\frac{p^{i+1}}{q^{i+1}}W_p(n) \tag{3.386}$$

$$= (1-\rho_m)W_q(0) + (1-\rho_m)\sum_{i=1}^{\infty}\frac{p^i}{q^i}W_q(n) + (1-\rho_m)\sum_{i=0}^{\infty}\frac{p^{i+1}}{q^{i+1}}W_p(n) \tag{3.387}$$

$$= (1-\rho_m)W_q(0) + (1-\rho_m)\sum_{i=0}^{\infty}\frac{p^{i+1}}{q^{i+1}}W_q(n+1) + (1-\rho_m)\sum_{i=0}^{\infty}\frac{p^{i+1}}{q^{i+1}}W_p(n) \tag{3.388}$$

$$= (1-\rho_m)W_q(0) + (1-\rho_m)\sum_{i=0}^{\infty}\frac{p^{i+1}}{q^{i+1}}(W_q(n+1)+W_p(n)) \tag{3.389}$$

$$= (1-\rho_m)W_q(0) \tag{3.390}$$

Then:

$$S = \sum_{m=1}^{\infty} P_m(1-\rho_m)W_q(0) = \sum_{m=1}^{\infty} P_m(1-\rho_m)\log_2\frac{q}{q(1-\rho_m)+p\rho_m} \tag{3.391}$$

$$V_0 = \frac{1}{C} \log_2 \frac{q}{q(1-\rho_0) + p\rho_0} \tag{3.392}$$

□

3.15 Extension to Arbitrary Initial Distributions

The proof of Thm. 1 only used the uniform input distribution to assert $U_i(0) = U_1(0)$ and replace $\mathsf{E}[U_i(0)]$ by $U_1(0) = \log_2(M-1)$ in equation (3.78). In Lemma 6, we have required that $U_i(0) < 0 \; \forall i$. However, even with uniform input distribution this is not the case when $\Omega = \{0, 1\}$. To avoid this requirement, the case where $\exists i : U_i(0) \geq 0$, and therefore $T^{(1)} = 0$, needs to be accounted for. Also, if $U_i(t) \geq C_2$, the probability that an initial fall back occurs is only upper bounded by p_f, which can be inferred from the proof of Lemma 4. Then, to obtain an upper bound on the expected stopping time $\mathsf{E}[\tau]$ for an arbitrary input distribution, it suffices to multiply the terms $\mathsf{E}[U_i(T^{(1)}) \mid T^{(1)} > 0, \theta = i] - U_i(0)$ in the proof of Lemma 6, equation (3.327), by the indicator $\mathbb{1}_{U_i(0)<0}$. Then, the bound on Lemma 6 becomes:

$$\sum_{i=1}^{M} \sum_{n=1}^{\infty} \mathsf{E}[U_i(t_0^{(n)} + T^{(n)}) - U_i(t_0^{(n)}) \,|\, \theta = i] \Pr(\theta = i) \leq C_2 \frac{p}{q} \frac{1 - \left(\frac{p}{q}\right)^N}{1 - \frac{p}{q}} + \mathsf{E}[(C_2 - U_i(0))\, \mathbb{1}_{U_i(0)<0}] \,. \tag{3.393}$$

By Thm. 3, the constant C_2 may be replaced with $q^{-1} \log_2(2q)$ in (3.393). Using the definition of p_f from (3.54) the following bound is obtained:

$$\mathsf{E}[T] \leq \mathsf{E}[T'] \leq 2^{-C_2} \frac{1 - 2^{-NC_2}}{1 - 2^{-C_2}} \frac{\log_2(2q)}{qC} + \mathsf{E}\left[\left(\frac{\log_2(2q)}{q} - U_i(0) \right) \frac{\mathbb{1}_{U_i(0)<0}}{C} \right] . \tag{3.394}$$

3.15.1 Generalized Achievability Bound

An upper bound on $\mathsf{E}[\tau]$, for an arbitrary initial distribution $\boldsymbol{\rho}_0$, can be derived using bound (3.394) on $\mathsf{E}[T]$, and bound (3.79) on $\mathsf{E}[\tau - T]$, to obtain:

$$\mathsf{E}[\tau] \leq \sum_{i=1}^{M} \left(\frac{\log_2\left(\frac{1-\rho_i(0)}{\rho_i(0)}\right)}{C} + \frac{\log_2(2q)}{q \cdot C} \right) \rho_i(0) \mathbb{1}_{\rho_0^{(i)} < 0.5} \tag{3.395}$$

$$+ \left\lceil \frac{\log_2(\frac{1-\epsilon}{\epsilon})}{C_2} \right\rceil \frac{C_2}{C_1} + \left(\frac{\log_2(2q)}{qC} - \frac{C_2}{C_1} \right) \frac{1 - \frac{\epsilon}{1-\epsilon} 2^{-C_2}}{1 - 2^{-C_2}} 2^{-C_2} \,. \tag{3.396}$$

For the special case where $\rho_i(0) \ll \frac{1}{2} \quad \forall i = 1, \ldots, M$, the log likelihood ratio can be approximated by $\log_2(\frac{\rho_i(0)}{1-\rho_i(0)}) \lessapprox \log_2(\rho_i(i))$ to obtain a simpler expression of the bound (3.396):

$$\mathsf{E}[\tau] < \frac{\mathcal{H}(\boldsymbol{\rho}_0)}{C} + \frac{\log_2(2q)}{q \cdot C} + \left\lceil \frac{\log_2(\frac{1-\epsilon}{\epsilon})}{C_2} \right\rceil \frac{C_2}{C_1} + \left(\frac{\log_2(2q)}{qC} - \frac{C_2}{C_1} \right) \frac{1 - \frac{\epsilon}{1-\epsilon} 2^{-C_2}}{1 - 2^{-C_2}} 2^{-C_2} \,, \tag{3.397}$$

where $\mathcal{H}(\boldsymbol{\rho}(0))$ is the entropy of the p.d.f. $\boldsymbol{\rho}_0$ in bits.

3.15.2 Uniform and Binomial Initial Distribution

Using the bound of equation (3.396), a better upper bound on the blocklength for a systematic encoder can be computed for uniform input distribution when $\Omega = \{0,1\}^K$. It can be shown that the systematic transmissions transform the uniform distribution into a binomial distribution, see [AYW20]. The bound is constructed by adding the K systematic transmissions to the bound in (3.396) applied to the binomial distribution as follows:

$$\mathsf{E}[\tau] \leq K + \sum_{i=0}^{K} \left\lceil \frac{\log_2(\frac{1-p^i q^{K-i}}{p^i q^{K-i}})}{C} + \frac{\log_2(2q)}{qC} \right\rceil \binom{K}{i} p^i q^{K-i} \mathbb{1}_{(q^{K-i}p^i < 0.5)}$$

$$+ \left\lceil \frac{\log_2(\frac{1-\epsilon}{\epsilon})}{C_2} \right\rceil \frac{C_2}{C_1} + \left(\frac{\log_2(2q)}{qC} - \frac{C_2}{C_1} \right) \frac{1 - \frac{\epsilon}{1-\epsilon} 2^{-C_2}}{1 - 2^{-C_2}} 2^{-C_2} \,. \tag{3.398}$$

This bound, which assumes SEAD and systematic transmission, is the tightest achievability bound that we have developed for the model.

Theorem 5. *[from [AYW20]] Suppose that* $\Omega = \{0,1\}^K$ *and* $\rho_i(0) = 2^{-K}\, \forall i \in \Omega$. *Then, for* $t = 1, \ldots, K$ *the partitioning rule* $S_0 = \{i \in \Omega \mid b_t^{(i)} = 0\}, S_1 = \{i \in \Omega \mid b_t^{(i)} = 1\}$, *results in systematic transmission:* $x^K = \theta$, *and achieves exactly equal partitioning* $P_0 = P_1 = \frac{1}{2}$.

Proof. First note that if $\Omega = \{0,1\}^K$, then for each $t = 1, \ldots, K$, exactly half of the items in $i \in \Omega$ have bit $b_t^{(i)} = 0$, and the other half have bit $b_t^{(i)} = 1$. The theorem holds for $t = 1$, since the partitioning $S_0 = \{i \in \Omega \mid b_1^{(i)} = 0\}, S_1 = \{i \in \Omega \mid b_1^{(i)} = 1\}$ results in half the messages in each partition, and all the messages have the same prior. For $t = 1, \ldots, K-1$ note the partitioning $S_0 = \{i \in \Omega \mid b_t^{(i)} = 0\}, S_1 = \{i \in \Omega \mid b_t^{(i)} = 1\}$ only considers the first t bits $b_1^{(i)}, \ldots, b_t^{(i)}$ of each message i. Thus, all item $\{j \in \Omega \mid b_1^{(j)}, \ldots, b_t^{(j)} = b_1, \ldots, b_t\}$ that share a prefix sequence $b_1, \ldots, b_t$ have shared the same partition at times $s = 1, \ldots, t$, and therefore share the same posterior. There are exactly 2^{K-t} such difference posteriors. Also, exactly half of the items that share the sequence $b_1, \ldots, b_t$ have bit $b_{t+1} = 0$ and are assigned to S_0 at time $t+1$, and the other half have bit $b_{t+1} = 1$ and are assigned to S_1 at time $t+1$. Then, S_0 and S_1 will each hold half the items in each posterior group at each next time $t+1$ for $t = 1, \ldots, K-1$, and therefore equal partitioning holds also at times $t = 2, \ldots, K$.

□

3.16 Achievability and Converse Bounds Graphs and Simulations

The achievability and converse bound in Thms. 1-5 are graphed in Fig. 3.5-3.8 vs. block-length τ, for channels with capacities $C = 0.90, 0.75, 0.50, 0.25$, and with decoding threshold $\epsilon = 10^{-3}$. The analytical bounds are validated with simulated rate results of the SEAD encoder implemented via the SPM algorithm with TOP partitioning shown in chapter 2. Fig. 3.5 shows all the curves for a channel with capacity $C = 0.50$. For this channel, the

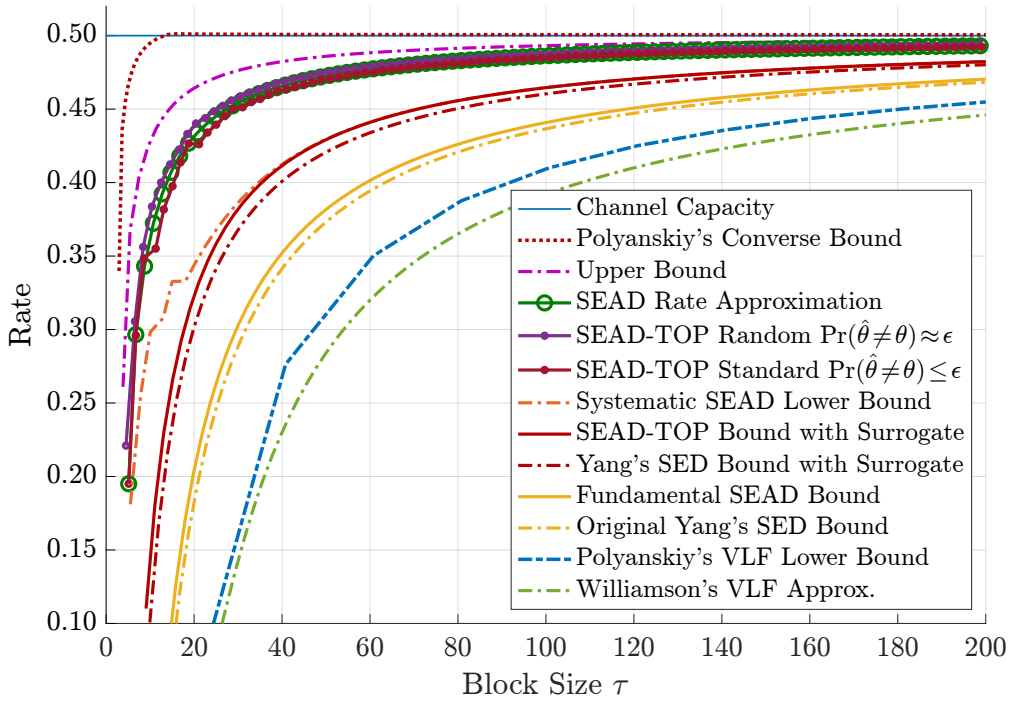

Figure 3.5: Achievability and converse rate bounds vs. blocklength τ over a channel with capacity $C = 0.50$, horizontal blue line. From top to bottom, the top red dotted curve $\cdot\cdot$ is Polyanskyi's converse bound [PPV11]; The magenta dash dot $-\cdot$ curve is the converse bound of Thm. 4; the solid green line $-\circ$ with circles is the approximation of the SEAD encoder described in equation (3.41); the solid purple and burgundy curves $-\bullet$ with circles are a the simulations from chapter 2 of the SEAD encoder with TOP partitioning with pseudo-random stopping rule, purple curve $-\bullet$, and with the standard rule (2.1), red curve $-\bullet$; the orange dash dot $-\cdot$ curve is the achievability bound of Thm. 5 for the systematic SPM algorithm. The bottom six curves are the curves described in Fig. 3.2. For all the curves, the decoding threshold is $\epsilon = 10^{-3}$.

converse bound of Thm. 4, see magenta curve labeled "Upper Bound" is below Polyanskiy's converse bound [PPV11], and the bound is validated by the simulations, with a rate performance below the bound. The approximation in equation (3.41), and described in the proof of Thm. 4, shown by the green solid line with circles, lies very close to the simulated rate, which shows that it is good approximation, when the channel's capacity is $C = 0.50$ and the decoding threshold $\epsilon = 10^{-3}$, this approximation is also very close when the channel's capacity changes to $C = 0.90$, $C = 0.75$, and $C = 0.25$, see Figs. 3.6-3.8.

The three achievability bounds introduced in this dissertation are shown in Fig. 3.5, for a

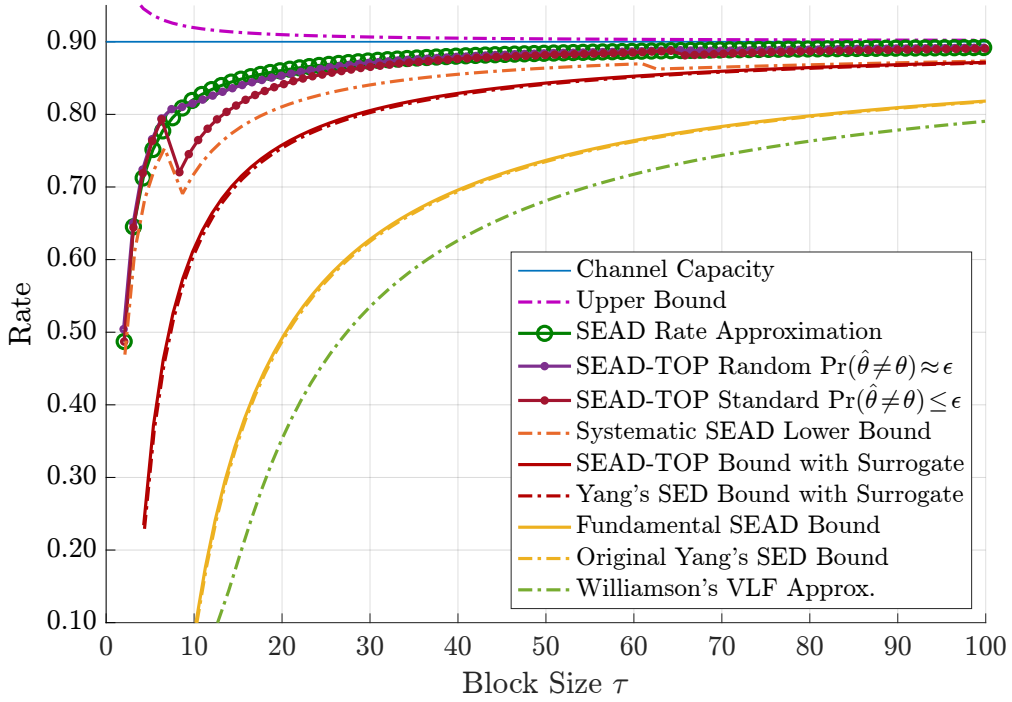

Figure 3.6: Achievability and Converse rate bounds vs. blocklength τ for a channel with capacity $C = 0.90$, horizontal blue line. These are the same curves shown in 3.5, but without Polyanskiy's converse bound and Polyanskiy's VLF lower bounds, and for a channel with higher capacity $C = 0.90$. For this channel, the converse bound of Thm. 4 is above the channel's capacity, the achievability bound of Thm. 5 for the systematic SPM algorithm, orange dash dot $-\cdot$ curve labeled "Systematic SEAD Lower Bound," closely approximates from below the simulation with standard stopping, solid burgundy line $-\bullet$ with circles labeled "SEAD-TOP Standard $\Pr(\hat{\theta} \neq \theta) \leq \epsilon$.

channel with capacity $C = 0.50$, and error probability $\Pr(\hat{\theta} \neq \theta)$ threshold $\epsilon = 10^{-3}$, and for channels with capacity $C = 0.90$, $C = 0.75$, and $C = 0.25$ in Fig. 3.6, 3.7, and 3.8. The highest bound is the bound of Thm. 5, for an encoder that enforces the SEAD constraints, when systematic transmission are used, and the input distribution is uniform over $\Omega = \{0, 1\}^K$, see orange dash dot line labeled "Systematic SEAD Lower Bound." The next highest bound is the lower bound (3.34) introduced in Thm. 3 for a system that enforces the SEAD constraints, see red solid curve labeled "SEAD-TOP Bound with Surrogate." This bound is a slight improvement over the SED lower bound by Yang *et al.* [YPA21] that is shown for comparison, see red dash-dot curve $-\cdot$ labeled "Yang's SED Bound with Surrogate." The fun-

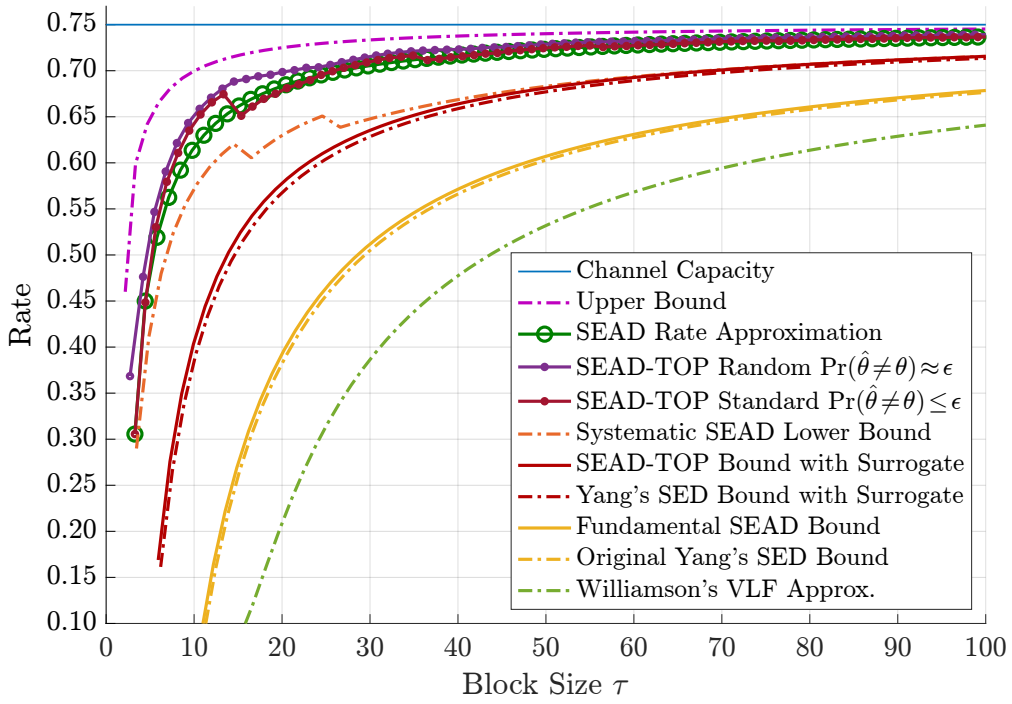

Figure 3.7: Achievability and Converse rate bounds vs. blocklength τ for a channel with capacity $C = 0.75$, horizontal blue line. These are the same curves in Fig. 3.6, but for a channel with capacity $C = 0.75$. Note that for this channel the converse becomes relevant, as it is below the channel's capacity.

damental achievability bound (3.27) of Thm. 1, without he surrogate process optimization, is shown to highlight the difference, see yellow solid line labeled "Fundamental SEAD Bound." This bound is also a slight improvement over Yang's original bound [YW19, YPA21], without the optimization of the surrogate martingale, see yellow dash dot $-\cdot$ line labeled "Original Yang's SED Bound." For comparison, Polyanskiy's lower bound [PPV11] for variable-length stop-feedback codes, labeled "Polyanskiy's VLF Lower Bound," is included in Fig. 3.5, and Williamnson's approximation [Wil14] of Polyanskiy's VLF bound is shown in Figs. 3.5-3.8. Since a stop feedback system is less capable than a full feedback system, a lower bound analyzed for the full feedback system is expected approaches capacity faster than the VLF bound, which is what the previous three bounds achieve.

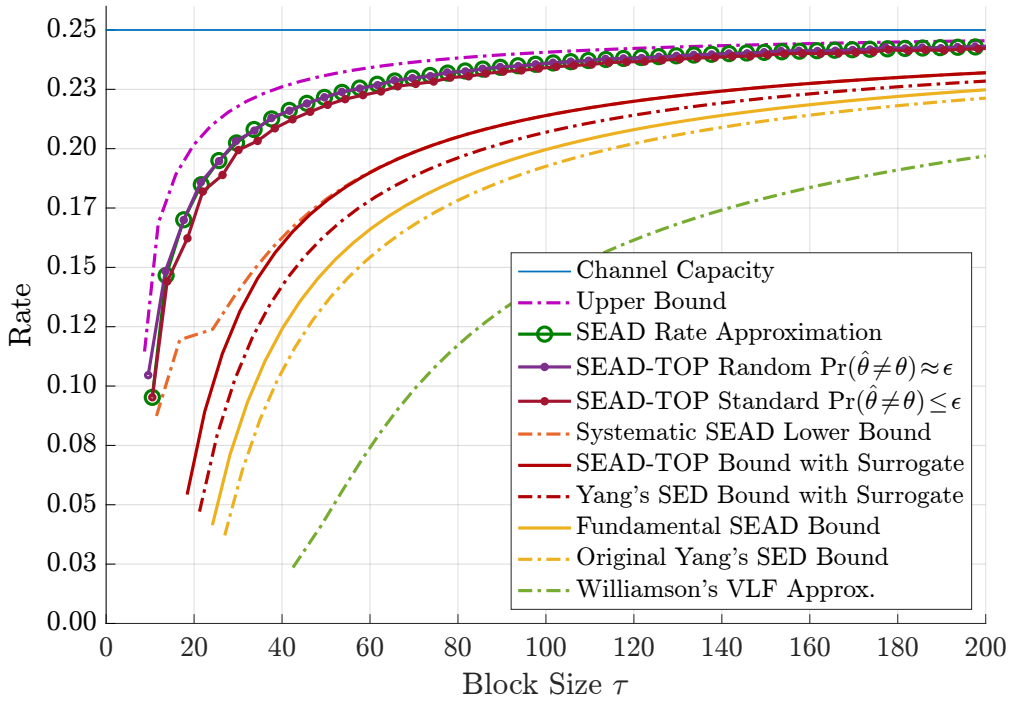

Figure 3.8: Achievability and Converse rate bounds vs. blocklength τ for a channel with capacity $C = 0.75$, horizontal blue line. These are the same curves in Fig. 3.6 and Fig. 3.7, but for a channel with capacity $C = 0.25$. Note that the converse bound is tighter and the achievability bounds are looser for this channel with lower capacity, compared to Figs. 3.5, 3.6, and 3.7.

3.17 Conclusions

Naghshvar *et al.* [NJW15] established the "small enough difference" (SED) rule for posterior matching partitioning, and used martingale theory to study asymptotic behavior, and then, also showed how to develop a non-asymptotic lower bound on the achievable rate. Yang *et al.* [YPA21] significantly improved the non-asymptotic achievable rate bound using martingale theory for the communication phase, and a Markov model for the confirmation phase, still maintaining the SED rule. However, partitioning algorithms that enforce the SED rule require a complex process of swapping messages back and forth between the two message sets S_0 and S_1, and updating the posteriors.

The fundamental Thm. of this section shows that the SEAD constraints suffice to achieve

at least the same lower bound that Yang *et al.* [YPA21] analyzed for more constrained SED encoder, before being optimized with a "surrogate martingale." The analysis forgoes martingale theory for the communication phase, in favor of a new method that requires looser encoding constraints. The new analysis necessitates a different definition of the communication phase time to include the time spent after possible events where the communication phase resumes after a fall back from the confirmation phase. The new communication phase time definition not only makes the analysis feasible, but also allows to slightly tighten the achievability bound. A significantly tighter bound was derived in this dissertation, that applies the analysis in the fundamental Thm. to an optimized "surrogate process" that lower bounds the decoding time of the original process. Moreover, the slight improvement, derived from the new definition of the communication phase time, is carried over to the bound analyzed with the "surrogate process," to yield an achievability bound higher than the bound optimized by a "surrogate martingale" by Yang *et al* [YPA21]. An initial systematic transmission phase, for which a tight analysis is possible, was used to obtain the highest achievability bounds in this dissertation.

A converse bound on the achievable rate is analyzed in this dissertation, for a system that decodes with an error probability bounded by a small threshold. The converse bound used the communication phase analysis introduced in the fundamental Thm., and separately analyzes the time spent in correct and wrong confirmation phases. Also, this dissertation approximates the performance of the SED and SEAD encoders by an expression that combines elements from the achievability and the converse bounds.

The simulation results obtained with the methods of chapter 2 validate the analytical results in this chapter, and show that, for some channels, the new analytical bounds bridge most of the rate gap between the previous achievability bounds and the rate from the simulated.

3.18 Proof of claim 2

Proof that $\mid U_i(t+1) - U_i(t) \mid = C_2$ is equivalent to $S_0 = \{j\}$ (or $S_1 = \{j\}$). First let's prove the converse, if the set containing j is not singleton, then constraint (3.25) does not hold. Without loss of generality, assume $j \in S_0$, and suppose $\exists l \neq j$ s.t. $l \in S_0$. Since $P_0 \geq \rho_j(y^t) + \rho_l(t)$, then, $\Delta = 2P_0 - 1 \geq 2\rho_i(y^t) + 2\rho_j(y^t) - 1 \geq 2\rho_j(y^t) - 1 > 0$. By equation (3.88), when $j \in S_y$, then:

$$
\begin{aligned}
U_j(t+1) - U_j(t) &= \log_2(2q) - \log_2\left(1 + (q-p)\frac{\Delta - \rho_j(y^t)}{1-\rho_j(y^t)}\right) && (3.399)\\
&\leq \log_2(2q) - \log_2\left(1 + (q-p)\frac{2\rho_j(y^t) + 2\rho_l(t) - 1 - \rho_j(y^t)}{1-\rho_j(y^t)}\right)\\
&= \log_2(2q) - \log_2\left(1 - (q-p) + (q-p)\frac{2\rho_l(t)}{1-\rho_j(y^t)}\right)\\
&< \log_2(2q) - \log_2(1-(q-p)) = C_2\,. && (3.400)
\end{aligned}
$$

Note from equation (3.399) that $U_j(t+1) - U_j(t)$ decreases with Δ, therefore, replacing Δ with a lower bound gives an upper bound of difference (3.399). For a lower bound, note that $\Delta \leq 1$, and that setting $\Delta = 1$ in (3.399), results in $U_j(t+1) - U_j(t) = 0$. In the case where $Y_{t+1} = X_{t+1} \oplus 1$, or $j \in S_{y^c}$, by equation (3.88), the difference $U_j(t+1) - U_j(t)$ is:

$$
\begin{aligned}
U_j(t+1) - U_j(t) &= \log_2(2p) - \log_2\left(1 - (q-p)\frac{\Delta - \rho_j(y^t)}{1-\rho_j(y^t)}\right) && (3.401)\\
&\geq \log_2(2p) - \log_2\left(1 - (q-p)\frac{2\rho_j(y^t) + 2\rho_l(t) - 1 - \rho_j(y^t)}{1-\rho_j(y^t)}\right) && (3.402)\\
&= \log_2(2p) - \log_2\left(1 + (q-p)\left(1 - \frac{2\rho_l(t)}{1-\rho_j(y^t)}\right)\right) && (3.403)\\
&> \log_2(2p) - \log_2(2q) = -C_2\,. && (3.404)
\end{aligned}
$$

To prove that if the set containing j is singleton, then $|U_j(t+1) - U_j(t)| = C_2$, note that $S_0 = \{j\} \implies \Delta = 2\rho_j(y^t) - 1$. The inequalities, therefore, become equalities and equations (3.399) and (3.401) become C_2 and $-C_2$ respectively.

3.19 Proof of existence of $U_i'(t)$ in Thm. 3

The proof that a process like the one described in Thm. 3 exists, consists of constructing one such process. Define the process $U_i'(t)$ by $U_i'(t) = U_i(t)$ when $Y^t \in \mathcal{B}^{\epsilon}_{(i,n)}$, for some $n \in \mathrm{N}$, otherwise $U_i'(t+1) = U_i'(t) + W_i'(t+1)$, where $W_i'(t+1)$ is a weight update to be defined, or just $C_2(\mathbb{1}_{(Y_{t+1}=0)} - \mathbb{1}_{(Y_{t+1}=1)})$ if $Y^t \notin \mathcal{Y}^{\epsilon}_{(i,n)}$. The update $W_i'(t+1)$ needs to meet at the same time the constraints (3.22), (3.24), (3.26), and (3.25) of Thm. 1, and the constraints of Thm. 2 with $B = C$.

Denote the transmitted symbol X_{t+1} by X, and the received symbol Y_{t+1} by Y, and let $X^c = X \oplus 1$. The symbol Y could either be X or X^c, and a message i could be in either set: $\{i \in S_0\}$ or $\{i \in S_1\}$. These cases combine to four distinguishable events. Define $W_i(t+1) \triangleq U_i(t+1) - U_i(t)$, and note that $W_i(t+1)$ can be derived from equation (3.88) as follows:

$$W_i(t+1) = \begin{cases} \log_2(2q) + a_i & \text{if } i \in S_0, Y = X \\ \log_2(2p) + b_i & \text{if } i \in S_0, Y = X^c \\ \log_2(2p) + c_i & \text{if } i \in S_1, Y = X^c \\ \log_2(2q) + d_i & \text{if } i \in S_1, Y = X \end{cases}, \tag{3.405}$$

where a_i, b_i, c_i and d_i are given by:

$$a_i = -\log_2\left(1 - (q-p)\frac{\rho_j(y^t) - \Delta}{1 - \rho_j(y^t)}\right) \tag{3.406}$$

$$b_i = -\log_2\left(1 + (q-p)\frac{\rho_j(y^t) - \Delta}{1 - \rho_j(y^t)}\right) \tag{3.407}$$

$$c_i = -\log_2\left(1 + (q-p)\frac{\rho_j(y^t) + \Delta}{1 - \rho_j(y^t)}\right) \tag{3.408}$$

$$d_i = -\log_2\left(1 - (q-p)\frac{\rho_j(y^t) + \Delta}{1 - \rho_j(y^t)}\right). \tag{3.409}$$

Let a'_i and d'_i be defined by:

$$a'_i \triangleq \mathbb{1}_{(\Delta<0)} \frac{1-\Delta}{1+\Delta} \log_2 \left(1-(q-p)\Delta\right) - \frac{p}{q} b_i, \tag{3.410}$$

$$d'_i \triangleq \mathbb{1}_{(d_i<0)} d_i - \mathbb{1}_{(d_i \geq 0)} \frac{p}{q} c_i \,, \tag{3.411}$$

then, define the update $W'_i(t+1)$ by:

$$W'_i(t+1) = \begin{cases} \log_2(2q) + a'_i & \text{if } i \in S_0, Y = X \\ \log_2(2p) + b_i & \text{if } i \in S_0, Y = X^c \\ \log_2(2p) + c_i & \text{if } i \in S_1, Y = X^c \\ \log_2(2q) + d'_i & \text{if } i \in S_1, Y = X \end{cases} . \tag{3.412}$$

Need to show that the constraints (3.26)-(3.25) of Thm. 1, and constraints (3.28) and (3.30) are satisfied. When $U'_i(t) \geq 0$, since $W'_i(t+1)$ is defined in the same manner as $W_i(t+1)$, then constraints (3.24) and (3.25) are satisfied.

The proof that $U'_i(t)$ satisfies constraints (3.26), (3.22), and (3.28) is split into the case where $\Delta \geq 0$, and the case where $\Delta < 0$.

3.20 Proof that $U'_i(t)$ satisfies (3.26), (3.22), and (3.28) when $\Delta \geq 0$.

It suffices to show that for all $y^t \in \mathcal{Y}^{\epsilon}_{(i,n)}$ and for all $i = 1, \ldots, M$, if $\Delta \geq 0$, then $\mathsf{E}[W'_i(t+1) \mid \theta = i, Y^t = y^t] = C$, Since $C > 0$ (constraint (3.22)), and any weighted average would add up to C (constraint (3.26)).

When $\Delta \geq 0$, then $\rho_i(y^t) + \Delta > 0$, and $a'_i = -\frac{p}{q} b_j$ since $d_i > 0$. The expectation $\mathsf{E}[W'_j(t+1) \mid \theta = i, Y^t = y^t]$ can be computed from (3.88), where ι_i depends on whether

$i \in S_0$ or $i \in S_1$. The expectation is given by either (3.413) or (3.414) respectively:

$$q \log_2(2q) - pb_i + p \log_2(2p) + pb_i = C \text{ if } i \in S_0 \tag{3.413}$$

$$q \log_2(2q) - pc_i + p \log_2(2p) + pc_i = C \text{ if } i \in S_1 \,. \tag{3.414}$$

This proofs constraints (3.26) and (3.22) satisfied. To proof that constraint (3.28) is satisfied, need to show that $W_i'(t+1) \leq W_i(t+1)$. If suffices to compare the cases where $W_i'(t+1) \neq W_i(t+1)$, that is, when $Y = X$. Need to show that $a_i' \leq a_i$ and $d_i' \leq d_i$. For this comparison, express a_i and d_i as positive logarithms as follows:

$$a_i = \log_2 \left(1 + \frac{(q-p)(\rho_i(y^t) - \Delta)}{1 - \rho_i(y^t) - (q-p)(\rho_i(y^t) - \Delta)} \right) \tag{3.415}$$

$$a_i' = \frac{p}{q} \log_2 \left(1 + \frac{(q-p)(\rho_i(y^t) - \Delta)}{1 - \rho_i(y^t)} \right) \tag{3.416}$$

$$d_i = \log_2 \left(1 + \frac{(q-p)(\Delta + \rho_i(y^t))}{1 - \rho_i(y^t) - (q-p)(\Delta + \rho_i(y^t))} \right) \tag{3.417}$$

$$d_i' = \frac{p}{q} \log_2 \left(1 + \frac{(q-p)(\rho_i(y^t) + \Delta)}{1 - \rho_i(y^t)} \right) . \tag{3.418}$$

Since $p \leq \frac{1}{2} \rightarrow \frac{p}{q} \leq 1$, then, it suffices to show that the argument of the logarithm in (3.415) is greater than that of (3.416), and similarly for (3.417) and (3.418). Furthermore, since all arguments share the term 1, with inequalities (3.419) and (3.420), shown below, suffices:

$$\frac{(q-p)(\rho_i(y^t) - \Delta)}{1 - \rho_i(y^t) - (q-p)(\rho_i(y^t) - \Delta)} \geq \frac{(q-p)(\rho_i(y^t) - \Delta)}{1 - \rho_i(y^t)} \tag{3.419}$$

$$\frac{(q-p)(\Delta + \rho_i(y^t))}{1 - \rho_i(y^t) - (q-p)(\rho_i(y^t) + \Delta)} \geq \frac{(q-p)(\rho_i(y^t) + \Delta)}{1 - \rho_i(y^t)} \,. \tag{3.420}$$

The numerators on both inequalities are the same, and positive, since $q - p > 0$ and $i \in S_0 \implies \rho_i(y^t) - \Delta \geq 0$. Also, $\Delta \geq 0 \implies \rho_i(y^t) + \Delta \geq 0$, thus, both denominators on the left hand side are smaller than those in the right side. Since the numerators are exactly the same, then, the inequalities hold.

3.20.1 Proof that $U_i'(t)$ satisfies (3.26), (3.22), and (3.28) when $\Delta < 0$.

Next, need to show that constraints (3.26), (3.22), and (3.28) are also satisfied when $\Delta < 0$. In the case where $\rho_i(y^t) + \Delta > 0$, then d_i' is still $-\frac{p}{q}c_i \leq d_i$ by equation (3.417). However, whenever $\Delta < 0$, the term $\frac{1-\Delta}{1+\Delta}\log_2(1-(q-p)\Delta) \geq 0$ is added to a_i'. To show that constraint (3.26) holds, recall from (3.92) that:

$$\sum_{i=1}^{M}\rho_i \mathsf{E}[W_i(t+1) \mid \theta=i, Y^t=y^t] - C = \sum_{i\in S_0}\rho_i(y^t)\,(qa_i+pb_i) + \sum_{i\in S_0}\rho_i(y^t)\,(qd_i+pc_i) \tag{3.421}$$

$$\geq \sum_{i\in S_0}\rho_i(y^t)\,(qa_i+pb_i) - \sum_{i\in S_1}\rho_i(y^t)\log_2\left(1-(q-p)^2\frac{\rho_i(y^t)-\alpha}{1-\rho_i(y^t)}\right) \tag{3.422}$$

$$\geq \sum_{i\in S_0}\rho_i(y^t)\,(qa_i+pb_i) - \frac{1+\alpha}{2}\log_2\left(1+(q-p)^2\alpha\right). \tag{3.423}$$

To obtain $\mathsf{E}[W_i'(t+1) \mid \mathcal{F}_t, \theta = i]$, replace a_i by a_i' in equation (3.423), and let $e_i \triangleq \frac{1+\alpha}{1-\alpha}\log_2(1+(q-p)^2\alpha)$. Then $a_i' = e_i - \frac{p}{q}b_i$ and $qa_i' + pb_i = qe_i$, and replace to obtain:

$$\sum_{i=1}^{M}\rho_i \mathsf{E}[W_i'(t+1) \mid \theta=i, Y^t=y^t] - C = \sum_{i\in S_0}\rho_i(y^t)qe_i + \sum_{i\in S_1}\rho_i(y^t)\,(qd_i+pc_i) \tag{3.424}$$

$$\geq qe_1\sum_{i\in S_0}\rho_i(y^t) - \frac{1+\alpha}{2}\log_2\left(1+(q-p)^2\alpha\right) \tag{3.425}$$

$$= \frac{1-\alpha}{2}e_1 - \frac{1+\alpha}{2}\log_2\left(1+(q-p)^2\alpha\right) \tag{3.426}$$

$$= \left(\frac{1-\alpha}{2}\frac{1+\alpha}{1-\alpha} - \frac{1+\alpha}{2}\right)\log_2\left(1+(q-p)^2\alpha\right) \tag{3.427}$$

$$= 0\,. \tag{3.428}$$

To show that $\mathsf{E}[W_i'(t+1) \mid \theta = i, Y^t = y^t] \geq 0$, (constraint (3.22)), note that d_i' is either unchanged from d_i, or it takes the same value of the case when $\Delta \geq 0$, therefore, it holds for $i \in S_1$. For $i \in S_0$. Note from the first term of equation (3.424), that $\mathsf{E}[W_i'(t+1) \mid \theta = i, Y^t = y^t] - 0 = \rho_i(y^t)qe_i$. Since $e_i \geq 0$, then the expectation is either C or greater.

Need to show that $W_i'(t+1) \leq W_i(t+1)$ (constraint (3.28)). It suffices to show that $a_i' \leq a_i$ and $d_i' \leq d_i$. Again, since d_i' is either d_i or it takes the same value of the case when $\Delta \geq 0$, it is only necessary that the inequality $a_i' = e_i - \frac{p}{q}b_i \leq a_i$ holds. It suffices to show that for a positive scalar γ:

$$\gamma\left(q\left(e_i - \frac{p}{q}b_i\right) + pb_i\right) \leq \gamma\left(qa_i + pb_i\right) . \tag{3.429}$$

When $\Delta < 0$, then $e_i > 0$, leading to the following equality:

$$q\left(e_i - \frac{p}{q}b_i\right) + pb_i = \frac{1+\alpha}{1-\alpha}\log_2\left(1 + (q-p)^2\alpha\right) . \tag{3.430}$$

Recall from equation (3.96) that:

$$qa_i + pb_i \geq -\log_2\left(1 - (q-p)^2\frac{\rho_{\min}+\alpha}{1-\rho_{\min}}\right) , \tag{3.431}$$

and let $\gamma = \frac{1-\alpha}{2}$, then, the scaled difference between the left and right terms in (3.429) is:

$$\begin{aligned}
&\frac{1-\alpha}{2}(qa_i + pb_i) - \frac{1+\alpha}{2}\log_2(1 + (q-p)^2\alpha) && (3.432)\\
&\quad \geq -\frac{1-\alpha}{2}\log_2\left(1-(q-p)^2\frac{\rho_{\min}+\alpha}{1-\rho_{\min}}\right) - \frac{1+\alpha}{2}\log_2\left(1-(q-p)^2\alpha\frac{\rho_{\min}-1}{1-\rho_{\min}}\right) && (3.433)\\
&\quad \geq -\log_2\left(1-\frac{(q-p)^2}{1-\rho_{\min}}\left(\rho_{\min}\frac{1+\alpha^2}{2} - \alpha^2\right)\right) && (3.434)\\
&\quad \geq -\log_2\left(1-\frac{(q-p)^2}{1-\rho_{\min}}\left(\rho_{\min}\frac{1+\alpha^2}{2} - \rho_{\min}\alpha\right)\right) && (3.435)\\
&\quad = -\log_2\left(1-\frac{(q-p)^2}{1-\rho_{\min}}\left(\rho_{\min}\frac{(1-\alpha)^2}{2}\right)\right) \geq 0 . && (3.436)
\end{aligned}$$

Equation (3.433) follows from (3.431). In (3.434), Jensen's inequality is used, where: $\frac{1-\alpha}{2}(\rho_{\min}+\alpha) + \frac{1+\alpha}{2}\alpha(\rho_{\min}-1) = -\alpha^2 + \rho_{\min}\frac{1-\alpha+\alpha+\alpha^2}{2}$. In (3.435) note that $\alpha \leq \rho_{\min} \implies \alpha^2 \leq \rho_{\min}\alpha$. Finally $1-2\alpha+\alpha^2 = (1-\alpha)^2 \geq 0$. This proves that $e_i - \frac{p}{q}b_i \leq a_i$, and therefore $W_i'(t+1) \leq W_i(t+1)$.

3.20.2 Proof that $U_i'(t)$ satisfies constraint (3.30)

Finally need to show that constraint (3.30) is satisfied, that is: $U_i'(T_{n+1}) - \frac{p}{q}(U_i(T_n) - C_2) \leq \frac{1}{q}\log_2(2q)$. It has already been shown that the update $W_i'(t)$ allows the process $U_i'(t)$ to meet constraints (3.22)-(3.25) of Thm. (1), and constraints (3.28) and (3.28). Note that the definition of $U_i'(t)$, in Thm. 2, guarantees that $U_i'(t)$ resets to the value of $U_i(t)$ at any time $t_0^{(n+1)}$, when $U_i(t)$ falls from confirmation, even if $U_i'(t)$ has not entered confirmation, where $U_i'(t) \geq 0$. It is straightforward to construct a third process $U_i''(t)$ that preserves all the properties of $U_i'(t)$, and with $U_i''(t) \geq 0$ if $U_i(t) \geq 0$. The process $U_i''(t)$ could be initialized by $U_i''(t_0^{(n)}) = U_i(t_0^{(n)})$. Then, it could be defined by $U_i''(t+1) = U_i''(t) + W_i''(t+1)$, where the step size $W_i''(t)$ is defined by: $W_i''(t+1) = \max\{\min\{W_i(t+1), -U_i(t)\}, W_i'(t+1)\}$. The inner minimum guarantees that $U_i''(t)$ reaches 0 if $U_i(t)$ does, and the outer maximum guarantees that the step size is at least that of $U_i'(t)$. Then, both processes $U_i(t)$ and $U_i''(t)$ cross 0, and enter confirmation at the same time, and share the same values when $U_i(t) < 0$, that is:

$$U_i(t+1) \geq 0 \quad \Longrightarrow \quad U_i''(t+1) \geq 0 \tag{3.437}$$

$$U_i(t) \leq 0 \quad \Longrightarrow \quad U_i''(t) = U_i(t)\,. \tag{3.438}$$

Using the process $U_i''(t)$ and equation (3.438), the expression in constraint (3.30) becomes:

$$U_i''(t+1) - \frac{p}{q}\left(U_i(t+1) - C_2\right) U_i(t)\left(1 - \frac{p}{q}\right) + W_i''(t+1) - \frac{p}{q}\left(W_i(t+1) - C_2\right). \tag{3.439}$$

In the case where $W_i''(t+1) = -U_i(t)$, then $U_i''(t+1) = 0$ and $U_i(t+1) \in [0, C_2]$, and:

$$U_i''(t+1) - \frac{p}{q}(U_i(t+1) - C_2) = -\frac{p}{q}(U_i(t) + W_i(t+1) - C_2) \tag{3.440}$$

$$= \frac{p}{q}C_2 - \frac{p}{q}(U_i(t) + W_i(t+1)) \tag{3.441}$$

$$\leq \frac{p}{q}C_2 \leq \frac{1}{q}\log_2(2q)\,. \tag{3.442}$$

The first inequality in (3.442) follows since $U_i''(t) = 0 \implies W_i(t+1) \geq -U_i(t)$, and the second inequality holds because:

$$\log_2(2q) - pC_2 = 1 + (1-p)\log_2(1-p) + p\log_2(p) = C \geq 0\,.$$

A constrained maximization of expression (3.439) is used for the case where $W_i''(t) > -U_i(t)$, where the constraint is $U_i(t) < 0$ (or $\rho_i(y^t) < \frac{1}{2}$). For simplicity, the constant $\frac{1}{q}\log_2(2q)$ from (3.439) is subtracted first. Let $i \in \{1, \ldots, M\}$ be arbitrary, and let $\rho \triangleq \rho_i(y^t)$, and $\alpha \triangleq |\Delta|$. Then, write $U_i''(t+1) - \frac{p}{q}(U_i(t+1) - C_2)$ in terms of ρ, p, and α, as a function $g(\rho, p, \alpha)$, using the definitions of $W_i(t)$ and $W_i''(t)$ in (3.405), (3.412) and (3.410)-(3.411). When $\{i \in S_0\} \cap \{\Delta < 0\}$ or $\{i \in S_1\} \cap \{\Delta \geq 0\}$, then $g(\rho, p, \alpha) = g_1(\rho, p, \alpha)$, given by:

$$g_1(\rho, p, \alpha) = \log_2\left(\frac{\rho}{1-\rho}\right)\left(1 - \frac{p}{q}\right) - \frac{p}{q}\log_2(2q) - \frac{p}{q}\log_2(2p) \tag{3.443}$$
$$+ \mathbb{1}_{\Delta<0}\frac{1+\alpha}{1-\alpha}\log_2(1 + (q-p)^2\alpha)$$
$$+ \frac{p}{q}\log_2\left(1 + (q-p)\frac{\rho+\alpha}{1-\rho}\right) + \frac{p}{q}\log_2\left(1 - (q-p)\frac{\rho+\alpha}{1-\rho}\right)\,,$$

and when $\{i \in S_0\} \cap \{\Delta \geq 0\}$ or $\{i \in S_1\} \cap \{\Delta < 0\}$, then $g(\rho, p, \alpha) = g_2(\rho, p, \alpha)$ given by:

$$g_2(\rho, p, \alpha) = \log_2\left(\frac{\rho}{1-\rho}\right)\left(1 - \frac{p}{q}\right) - \frac{p}{q}\log_2(2q) - \frac{p}{q}\log_2(2p) \tag{3.444}$$
$$+ \frac{p}{q}\log_2\left(1 + (q-p)\frac{\rho-\alpha}{1-\rho}\right) + \frac{p}{q}\log_2\left(1 - (q-p)\frac{\rho-\alpha}{1-\rho}\right)\,.$$

3.20.3 Maximizing $g_1(\rho, p, \alpha)$, from (3.443)

The maximum of (3.443) happens when $\Delta < 0$, since the term with the indicator function is non-negative. Since $\alpha \leq \frac{1}{3}$, then $\frac{1+\alpha}{1-\alpha} \leq 2$, which is used to write a function $f(\rho, \alpha)$ that upper bounds $g_1(\rho, p, \alpha)$, that is: $g_1(\rho, p, \alpha) \leq f(\rho, \alpha)$, given by:

$$f(\rho,\alpha) \triangleq \log_2\left(\frac{\rho}{1-\rho}\right)\left(1-\frac{p}{q}\right) + 2\log_2(1+(q-p)^2\alpha) - \frac{p}{q}\log_2(2q) - \frac{p}{q}\log_2(2p) \\ + \frac{p}{q}\log_2\left(1+(q-p)\frac{\rho+\alpha}{1-\rho}\right) + \frac{p}{q}\log_2\left(1-(q-p)\frac{\rho+\alpha}{1-\rho}\right) . \tag{3.445}$$

To show that $g_1(\rho, p, \alpha) \leq \frac{1}{q}\log_2(2q)$, it suffices to show that of $f(\rho, \alpha) \leq \frac{1}{q}\log_2(2q)$, which is done by solving the following constrained maximization:

$$\textbf{maximize } f(\rho,\alpha) \tag{3.446}$$

$$\textbf{subject to } \rho \leq \frac{1}{2}, \alpha \leq 1-2\rho\,, \tag{3.447}$$

The procedure to find the maximum is to show that $f(\rho, \alpha)$ is increasing in ρ, by showing that $\frac{d}{d\rho}f \geq 0$, and then evaluate it at the maximum set by the constraints. Note that $\frac{d}{d\rho}\frac{\rho}{1-\rho} = \frac{1}{(1-\rho)^2}$ and $\frac{d}{d\rho}\frac{\rho+\alpha}{1-\rho} = \frac{1}{(1-\rho)^2} + \frac{\alpha}{(1-\rho)^2}$

$$\frac{d}{d\rho}\ln(2)f(\rho,\alpha) = \frac{q-p}{q}\frac{1}{(1-\rho)\rho} + \frac{p}{q}\frac{1+\alpha}{(1-\rho)^2}\left(\frac{(q-p)}{1+(q-p)\frac{\rho+\alpha}{1-\rho}} - \frac{(q-p)}{1-(q-p)\frac{\rho+\alpha}{1-\rho}}\right) . \tag{3.448}$$

Factor out the positive constant $\frac{1}{q}\frac{q-p}{1-\rho}$, to obtain:

$$\frac{q(q-p)}{1-\rho}\frac{d}{d\rho}\ln(2)f(\rho,\alpha) = \frac{1}{\rho} + p\frac{1+\alpha}{1-\rho}\left(\frac{1}{1+(q-p)\frac{\rho+\alpha}{1-\rho}} - \frac{1}{1-(q-p)\frac{\rho+\alpha}{1-\rho}}\right) \tag{3.449}$$

$$= \frac{1}{\rho} + p\frac{1+\alpha}{1-\rho}\frac{\left(1-(q-p)\frac{\rho+\alpha}{1-\rho}\right) - \left(1+(q-p)\frac{\rho+\alpha}{1-\rho}\right)}{1-(q-p)^2\left(\frac{\rho+\alpha}{1-\rho}\right)^2} \tag{3.450}$$

$$= \frac{1}{\rho} - 2\frac{p(1+\alpha)(q-p)(\rho+\alpha)}{(1-\rho)^2-(q-p)^2(\rho+\alpha)^2} \tag{3.451}$$

$$= \frac{(1-\rho)^2-(q-p)^2(\rho+\alpha)^2-2p\rho(1+\alpha)(q-p)(\rho+\alpha)}{\rho(1-\rho)^2-\rho(q-p)^2(\rho+\alpha)^2}\,. \tag{3.452}$$

It suffices to show that the top of equation (3.452) is non-negative. Since it decreases when $\alpha \le 1-2\rho$, then:

$$\begin{aligned}
&(1-\rho)^2-(q-p)^2(\rho+\alpha)^2 - 2p\rho(1+\alpha)(q-p)(\rho+\alpha)\\
&\quad\ge (1-\rho)^2 - (q-p)^2(\rho+1-2\rho)^2 - 2p\rho(1+1-2\rho)(q-p)(\rho+1-2\rho)\\
&\quad= (1-\rho)^2 - (q-p)^2(1-\rho)^2 - 2p\rho(2-2\rho)(q-p)(1-\rho)\\
&\quad= (1-\rho)^2(1-(q-p)^2 - (q-p)4p\rho)\\
&\quad= (1-\rho)^2 4p(q-\rho(q-p)) > 4p(1-\rho)^2(q-\rho) > 0\,.
\end{aligned} \tag{3.453}$$

In equation (3.453), applies the following: $(q-p)^2 = (1-2p)^2 = 1-4p+4p^2 = 1-4pq$, and $\rho(q-p) < \rho q < \rho$. Since $\frac{\partial}{\partial\rho}f > 0$, then f is increasing in ρ, and thus ρ is replaced by $\frac{1-\alpha}{2}$ for an upper bound. Since $\frac{\rho+\alpha}{1-\rho} = \frac{\frac{1-\alpha}{2}+\alpha}{1-\frac{1-\alpha}{2}} = \frac{1-\alpha+2\alpha}{2-1+\alpha} = 1$, then $f(\alpha) \triangleq f\left(\frac{1-\alpha}{2},\alpha\right)$ is given by:

$$f(\alpha) = \log_2\left(\frac{1-\alpha}{1+\alpha}\right)\left(1-\frac{p}{q}\right) + 2\log_2(1+(q-p)^2\alpha) \tag{3.454}$$

$$-\frac{p}{q}\log_2(2q) - \frac{p}{q}\log_2(2p) + \frac{p}{q}\log_2(1+(q-p)) + \frac{p}{q}\log_2(1-(q-p)) \tag{3.455}$$

$$= \log_2\left(\frac{1-\alpha}{1+\alpha}\right)\left(1-\frac{p}{q}\right) + 2\log_2(1+(q-p)^2\alpha). \tag{3.456}$$

To complete the proof, it suffices to show that the last expression decreases in α:

$$\frac{d}{d\alpha}\ln(2)f(\alpha) = \left(1-\frac{p}{q}\right)\frac{1+\alpha}{1-\alpha}\frac{-(1+\alpha)-(1-\alpha)}{(1+\alpha)^2}+\frac{2(q-p)^2}{1+(q-p)^2\alpha} \tag{3.457}$$

$$= -2\frac{1}{q}\frac{q-p}{1-\alpha^2}+2\frac{(q-p)^2}{1+(q-p)^2\alpha} \tag{3.458}$$

$$=2(q-p)\left(-\frac{1}{q}\frac{1}{1-\alpha^2}+\frac{q-p}{1+(q-p)^2\alpha}\right) \tag{3.459}$$

$$\leq -2(q-p)\left(1+\frac{p}{q}-q+p\right) = -2p(q-p)\left(2+\frac{1}{q}\right) < 0\,. \tag{3.460}$$

Since f is decreasing, then the maximum of equation (3.443) is 0, at $\alpha = 0$.

3.20.4 Maximizing $g_2(\rho, p, \alpha)$ from (3.444)

First, write $g_2(\rho,p,\alpha) = f(\rho,\alpha) - \frac{p}{q}\log_2(2q) - \frac{p}{q}\log_2(2p)$, where $f(\rho,\alpha)$ is defined by:

$$f(\rho,\alpha) \triangleq \log_2\left(\tfrac{\rho}{1-\rho}\right)\left(1-\frac{p}{q}\right)+\frac{p}{q}\log_2\left(1+(q-p)\tfrac{\rho-\alpha}{1-\rho}\right)+\frac{p}{q}\log_2\left(1-(q-p)\tfrac{\rho-\alpha}{1-\rho}\right). \tag{3.461}$$

Since only $f(\rho,\alpha)$ depends on ρ, it suffices to solve the following constrained maximization:

$$\textbf{maximize} \qquad f(\rho,\alpha) \tag{3.462}$$

$$\textbf{subject to} \qquad \rho \leq \frac{1}{2}, \alpha \leq 1-2\rho\,. \tag{3.463}$$

Combining the last two terms we obtain:

$$f(\rho,\alpha) = \log_2\left(\frac{\rho}{1-\rho}\right)\left(1-\frac{p}{q}\right)+\frac{p}{q}\log_2\left(1-(q-p)^2\left(\frac{\rho-\alpha}{1-\rho}\right)^2\right). \tag{3.464}$$

The first term increases with ρ, and the second one decreases as the quotient $\left(\frac{\rho-\alpha}{1-\rho}\right)$ increases in absolute value. For $\rho \leq \frac{1}{3}$, it is possible to have $\rho = \alpha$, leaving only (3.464). However, for $\rho \geq \frac{1}{3}$, the quotient is positive because $\alpha \leq 1-2\rho \leq \frac{1}{3}$. The smallest value of the quotient

is then $\frac{1-\alpha}{1-\rho}\frac{3\rho-1}{1-\rho} = \frac{2\rho}{1-\rho} - 1$, with squared $1 - \frac{4\rho(1-2\rho)}{(1-\rho)^2}$. Let $f(\rho)$ be as defined in equation (3.465), then the maximum of $g(\rho, \alpha)$ is bounded by the maximum of $f(\rho)$, where:

$$f(\rho) \triangleq \log_2\left(\frac{\rho}{1-\rho}\right)\left(1-\frac{p}{q}\right) + \frac{p}{q}\log_2\left(1-(q-p)\frac{3\rho-1}{1-\rho}\right) + \frac{p}{q}\log_2\left(1+(q-p)\frac{3\rho-1}{1-\rho}\right). \quad (3.465)$$

The behavior of $f(\rho)$ is analyzed by taking the first derivative:

$$\frac{d}{d\rho}f(\rho) = \frac{q-p}{q\ln(2)}\frac{1-\rho}{\rho}\frac{1}{(1-\rho)^2} + \frac{p}{q\ln(2)}\left(\frac{(q-p)\frac{2}{(1-\rho)^2}}{1+(q-p)\frac{3\rho-1}{1-\rho}} - \frac{(q-p)\frac{2}{(1-\rho)^2}}{1-(q-p)\frac{3\rho-1}{1-\rho}}\right). \quad (3.466)$$

For convenience, scale the derivative by the positive term $\frac{(1-\rho)^2}{q-p}q\ln(2)$, to obtain:

$$\begin{aligned}
\frac{d}{d\rho}f(\rho)\frac{(1-\rho)^2}{q-p}q\ln(2) =& \frac{1-\rho}{\rho} - p\frac{2}{1-(q-p)\frac{3\rho-1}{1-\rho}} + p\frac{2}{1+(q-p)\frac{3\rho-1}{1-\rho}} && (3.467)\\
=& \frac{1-\rho}{\rho} + 2p\frac{1-(q-p)\frac{3\rho-1}{1-\rho} - 1 - (q-p)\frac{3\rho-1}{1-\rho}}{1-(q-p)\frac{3\rho-1}{1-\rho}} && (3.468)\\
=& \frac{1-\rho}{\rho} - 4p(q-p)\frac{\frac{3\rho-1}{1-\rho}}{1-(q-p)\frac{3\rho-1}{1-\rho}} && (3.469)\\
=& \frac{1-\rho}{\rho} - 4p(q-p)\frac{3\rho-1}{1-(q-p)(3\rho-1)} && (3.470)\\
=& \frac{1-\rho-(q-p)(3\rho-1)(1-\rho+4p\rho)}{\rho-(q-p)\rho(3\rho-1)}. && (3.471)
\end{aligned}$$

To show that $g' \geq 0$ in $[\frac{1}{3}, \frac{1}{2}]$, it suffices to show that the top of equation (3.471) is positive:

$$\begin{aligned}
1-\rho-(q-p)(3\rho-1)(1-\rho+4p\rho) \geq& 1 - \frac{1}{2} - (q-p)\left(\frac{3}{2}-1\right)(1-\rho(1-4p)) && (3.472)\\
=& \frac{1}{2} - \frac{1-2p}{2}(1-\rho(1-4p)) && (3.473)\\
=& \frac{1}{2} - \frac{1-2p}{2} + \frac{1-2p}{2}\rho(1-4p) && (3.474)\\
=& \frac{1}{2} - \frac{1}{2} + p + \rho\frac{(1-2p)(1-4p)}{2} && (3.475)\\
=& p + \rho\frac{(1-2p)(1-4p)}{2}. && (3.476)
\end{aligned}$$

When $p \leq \frac{1}{4}$, then $(1-4p) > 0$, and the second term is non-negative, therefore, the derivative is positive. When $p > \frac{1}{4}$, then $1 - 4p \geq -1$, $0 \leq 1 - 2p < \frac{1}{2}$, and:

$$p + \rho\frac{(1-2p)(1-4p)}{2} \geq \frac{1}{4} - \rho\frac{1}{4} = \frac{1-\rho}{4} \geq \frac{1}{8} > 0\,.$$

This shows that f is increasing in ρ, with maximum at $\rho = \frac{1}{2}$, and where $\frac{\rho}{1-\rho} = 1$. Then, the maximum of $f(\rho)$ is given by:

$$f\left(\tfrac{1}{2}\right) = \log_2(1)\left(1-\tfrac{p}{q}\right) + \frac{p}{q}\log_2(1+(q-p)) + \frac{p}{q}\ln(1-(q-p)) - \frac{p}{q}\log_2(2q) - \frac{p}{q}\log_2(2p) = 0\,.$$

This shows that the maximum of $f(\rho,\alpha) - \frac{p}{q}\log(2q) - \frac{p}{q}\log_2(2p)$ is zero, thus the maximum of $g_1(\rho,p,\alpha)$ is also zero. Since the maximum of $g(\rho,p,\alpha)$ is zero in both cases, $\Delta < 0$ and $\Delta \geq 0$, then $U_i'(t+1) - \frac{p}{q}\left(U_i(t+1) - C_2\right) \leq \frac{1}{q}\log_2(2q)$.

Finally, the last claim needs to be proved, which is that $B = \frac{1}{q}\log_2(2q)$ is the smallest value for a system that enforces the SED constraint. It suffices to note that the surrogate process described in [YPA21], Sec. V E is a strict martingale. A process $U_i'(t)$ with a lower B value would not comply with constraint (3.8), and therefore would also fail to meet constraint (3.26).

3.21 Proof: Confirmation Phase State Space 3

Proof. Suppose that for times $t = s$ and $t = s+1$, the partitioning is fixed at $S_0 = \{j\}$ and $S_1 = \Omega \setminus \{j\}$. Need to show that $\forall i \in \Omega$, if $Y_s = 0$ and $Y_{s+1} = 1$, then, $\rho_i(y^s) = \rho_i(y^{s+2})$. Using the update formula (3.85), at time $t = s+1$ the updated $\rho_j(y^{s+1})$, for $i \neq j$, is given

by:

$$\rho_i(y^{s+1}) = \frac{p\rho_i(y^s)}{q\rho_j(y^s)+p(1-\rho_j(y^s))} = \frac{p\rho_i(y^s)}{\rho_j(y^s)(q-p)+p} \tag{3.477}$$

$$\rho_j(y^{s+1}) = \frac{q\rho_j(y^s)}{q\rho_j(y^s)+p(1-\rho_j(y^s))} = \frac{q\rho_j(y^s)}{\rho_j(y^s)(q-p)-p}\,. \tag{3.478}$$

At time $t = s+2$, since $Y_{s+2} = 1$, equation (3.85) for $i \neq j$ results in:

$$\rho_i(y^{s+2}) = \frac{q\rho_i(y^{s+1})}{(p-q)\rho_j(y^{s+1})+q} = \frac{q\frac{p\rho_i(y^s)}{\rho_j(y^s)(q-p)+p}}{(p-q)\frac{q\rho_j(y^s)}{\rho_j(y^s)(q-p)+p}+q} \tag{3.479}$$

$$= \frac{qp\rho_i(y^s)}{(p-q)q\rho_j(y^s)+q(\rho_j(y^s)(q-p)+p)} \tag{3.480}$$

$$= \frac{qp\rho_i(y^s)}{-(q-p)q\rho_j(y^s)+(q-p)q\rho_j(y^s)+qp} = \frac{qp\rho_i(y^s)}{qp} = \rho_i(y^s)\,. \tag{3.481}$$

And for $i = j$ equation (3.85) results in:

$$\rho_j(y^{s+2}) = \frac{p\rho_j(y^{s+1})}{(p-q)\rho_j(y^{s+1})+q} = \frac{p\frac{q\rho_j(y^s)}{\rho_j(y^s)(q-p)+p}}{(p-q)\frac{q\rho_j(y^s)}{\rho_j(y^s)(q-p)+p}+q} \tag{3.482}$$

$$= \frac{pq\rho_j(y^s)}{(p-q)q\rho_j(y^s)+q(\rho_j(y^s)(q-p)+p)} \tag{3.483}$$

$$= \frac{qp\rho_j(y^s)}{-(q-p)q\rho_j(y^s)+(q-p)q\rho_j(y^s)+qp} = \frac{qp\rho_j(y^s)}{qp} = \rho_j(y^s)\,. \tag{3.484}$$

Then, for all $i\Omega$ each posterior at time $t = s+1$ is given by: $\rho_i(y^{s+2}) = \rho_i(y^s)$. The same equalities hold when $Y_{s+1} = 1$ and $Y_{s+2} = 0$, where the only difference is that p and q are interchanged. Finally, by induction, $\rho_i(y^{s+2r}) = \rho_i(y^s)$ for $r = 1, \ldots,$ if for every $t = s, \ldots, s+2r-1$ the partitions are fixed at $S_0 = \{j\}$ and $S_1 = \Omega \setminus \{j\}$, and $\sum_{k=1}^{2r} Y_{s+k} = 0$. □

3.22 Proof of Inequality (3.331), Chapter. 3

Proof. Need to show that the following inequality holds:

$$\mathsf{E}[U_i(T_n) \mid T^{(n+1)}>0, \theta=i] \geq \mathsf{E}[U_i(T_n) \mid T^{(n)}>0, \theta=i] \tag{3.485}$$

Recall that $\mathcal{C}(t_0^{(n)})$ is the event that message i enters confirmation after time $t_0^{(n)}$, rather than another message $j \neq i$ ending the process by attaining $U_j(t) \geq \log_2(1-\epsilon) - \log_2(\epsilon)$. This event is defined by $\mathcal{C}(t_0^{(n)}) \triangleq \{\exists t > t_0^{(n)} : U_i(t) \geq 0\}$. Using Bayes rule, the expectation $\mathsf{E}[U_i(T_n) \mid T^{(n)} > 0, \theta=i]$ can be expanded as a sum of expectations conditioned on events that are defined in terms of $\{T^{(n+1)} > 0\}$, $\{T^{(n)} > 0\}$ and $\mathcal{C}(t_0^{(n)})$, and whose union is the full event space to leave only the original conditioning $\{T^{(n)} > 0\}$. These events are $\mathcal{C}(t_0^{(n)}) \cap \{T^{(n+1)} > 0\}$, $\mathcal{C}(t_0^{(n)}) \cap \{T^{(n+1)} \leq 0\}$, $\neg\mathcal{C}(t_0^{(n)}) \cap \{T^{(n+1)} > 0\}$ and $\neg\mathcal{C}(t_0^{(n)}) \cap \{T^{(n+1)} \leq 0\}$. Note that $\neg\mathcal{C}(t_0^{(n)}) \implies \{T^{(n+1)} = 0\}$ and therefore, the third event vanishes. The expansion is given by:

$$\begin{aligned}
&\mathsf{E}[U_i(T_n) \mid T^{(n)}>0, \theta=i] = \\
&\quad \mathsf{E}[U_i(T_n) \,|\, T^{(n+1)} > 0, T^{(n)} > 0, \theta = i] \Pr(\mathcal{C}(t_0^{(n)}), T^{(n+1)}> 0 \,|\, T^{(n)} > 0, \theta=i) && (3.486)\\
&\quad + \mathsf{E}[U_i(T_n) \,|\, \mathcal{C}(t_0^{(n)}), T^{(n+1)}\leq 0, T^{(n)} > 0, \theta=i] \Pr(\mathcal{C}(t_0^{(n)}), T^{(n+1)} \leq 0 \,|\, T^{(n)} > 0, \theta=i) && (3.487)\\
&\quad + \mathsf{E}[U_i(T_n) | \neg\mathcal{C}(t_0^{(n)}), T^{(n+1)}\leq 0, T^{(n)} > 0, \theta=i] \Pr(\neg\mathcal{C}(t_0^{(n)}), T^{(n+1)}\leq 0 \,|\, T^{(n)} > 0, \theta=i). && (3.488)
\end{aligned}$$

Since $\{T^{(n+1)} > 0\} \implies \mathcal{C}(t_0^{(n)}) \cap \{T^{(n)} > 0\}$, we can omit the conditioning on $\mathcal{C}(t_0^{(n)})$ and $\{T^{(n)} > 0\}$ when accompanied by $\{T^{(n+1)} > 0\}$. By the independence of the confirmation phase from the crossing value $U_i(T_n)$ derived from the fix state count of the Markov Chain we have that:

$$\mathsf{E}[U_i(T_n) \mid \mathcal{C}(t_0^{(n)}), T^{(n+1)} \leq 0, T^{(n)} > 0, \theta=i] = \mathsf{E}[U_i(T_n) \mid T^{(n+1)} > 0, T^{(n)} > 0, \theta=i]\,. \tag{3.489}$$

Therefore, the expectation in (3.487) can be replaced by the one in (3.486). Then, the probabilities in (3.486) and (3.487) are added, to obtain $\Pr(\mathcal{C}(t_0^{(n)}) \mid T^{(n)} > 0, \theta = i)$. Note that $\neg\mathcal{C}(t_0^{(n)}) \implies \{T^{(n+1)} \leq 0\}$, thus the conditioning on $\{T^{(n+1)} \leq 0\}$ is redundant with $\neg\mathcal{C}(t_0^{(n)})$. Then, the expectation in the left of (3.486) is also given by:

$$\mathsf{E}[U_i(T_n) \mid T^{(n)} > 0, \theta = i] = \mathsf{E}[U_i(T_n) \mid T^{(n+1)} > 0, \theta = i] \Pr(\mathcal{C}(t_0^{(n)}) \mid T^{(n)} > 0, \theta = i) \tag{3.490}$$

$$+ \mathsf{E}[U_i(T_n) \mid \neg\mathcal{C}(t_0^{(n)}), T^{(n)} > 0, \theta = i] \Pr(\neg\mathcal{C}(t_0^{(n)}) \mid T^{(n)} > 0, \theta = i) \tag{3.491}$$

The event $\neg\mathcal{C}(t_0^{(n)}) \cap \{T^{(n)} > 0\} \cap \{\theta = i\}$ implies that the process decodes in error at the n^{th} communication phase round, which results in $U_i(T_n) < 0$. Therefore, we have that $\mathsf{E}[U_i(T_n) \mid \neg\mathcal{C}(t_0^{(n)}), T^{(n+1)} \leq 0, \theta = i] < 0$, This makes the left side of (3.490) an average of the positive quantity in the right of (3.490) and the negative quantity in (3.491). Then:

$$\mathsf{E}[U_i(T_n) \mid T^{(n)} > 0, \theta = i] \leq \mathsf{E}[U_i(T_n) \mid T^{(n+1)} > 0, \theta = i] \Pr(\mathcal{C}(t_0^{(n)}) \mid T^{(n)} > 0, \theta = i) \tag{3.492}$$

$$\leq \mathsf{E}[U_i(T_n) \mid T^{(n+1)} > 0, \theta = i] \tag{3.493}$$

The last inequality (3.493) follows because the expectation is positive, and is multiplied by a probability, $0 \leq \Pr(\mathcal{C}(t_0^{(n)}) \mid T^{(n)} > 0, \theta = i) \leq 1$, in (3.492). □

CHAPTER 4

Causal Encoding

4.1 Introduction

Consider a feedback communication system over the BSC where the information source generates the information sequence as the transmission progresses, instead of making it available to the encoder before the start of the transmission. The information sequence is generated at the source symbol rate of λ [bits/sec], and the transmitter sends coded bits through the BSC at a rate μ [bits/sec] (see the model of Fig. 4.1). The decoder needs to produce an estimate of the information sequence promptly and reliably, so it can be used by a time-sensitive application, for example, a control application like stabilizing an unstable plant over a noisy link (see [SH16]), the remote teleoperation of an aircraft or robot (see [AJB10, AJD13]), and in sequential optimization of the beamforming vectors during the initial access phase of communication over mmWave (see [CRJ19]). In the causal model, the time t indexes the channel symbols, not unlike the models of chapters 2 and 3. The decoding time T_d, also in channel symbols, is the time the decoder takes to estimate the information sequence, starting from the time when the first information bit arrives at the encoder. The "causality constraint" restricts the encoder to encode each symbol X_t, using only the information sequence symbols and the feedback symbols available at time t. The "causal constraint" allows to use every previous feedback symbol $Y_1, Y_2, \ldots, Y_{t-1}$, but for values of t for which $\lfloor \frac{t\lambda}{\mu} \rfloor < K$, only the first $\lfloor \frac{t\lambda}{\mu} \rfloor$ bits of the information sequence are available to the encoder.

The goal of "causal encoding" is to design a system, an encoder and a matched decoder, that encodes channel bits "causally" and decodes the information sequence at the earliest possible time. The encoding "causality" consists of using only the information sequence symbols "causally available" from the "streaming source" at each channel symbol time t. In "causal encoding" every opportunity to transmit a symbol over the channel is counted in the time T_d, and the encoder needs to make the most effective use of every channel symbol to reduce the expected decoding time $\mathsf{E}[T_d]$. This research targets blocklengths from a few tens of bits to a few thousand bits, the same range targeted in the non-causal setting of chapters 2 and 3, where variable-length coding is required for the rate to approach capacity with low FER. This dissertation proposes variable-length, causal encoding functions that seek to decode with the smallest expected decoding time $\mathsf{E}[T_d]$, and satisfy a reliability target similar to that of Sec. 2 and Sec. 3, and maintaining a relatively low decoding complexity.

4.2 Background

Causal Encoding has been analyzed before with feedback. Sahai [Sah08] studied fixed delay "anytime" codes, and extended the study to the case where the feedback is received with a small delay. Gorantla and Coleman [GC11] introduced a new class of causal coding problems to optimize sequential decision-making in systems with feedback and proved the existence of of an optimal stretegy for a fixed cost function. Lalitha *et al.* [LKJ20] proposed a causal encoding version of Horstein's scheme over the BSC. Causal encoding without feedback has also been studied as Streaming Transmission in a related setting where decoding of each block is subject to a fix delay deadline [KD14, LTK15]. An error, or erasure, is declared when the deadline is missed.

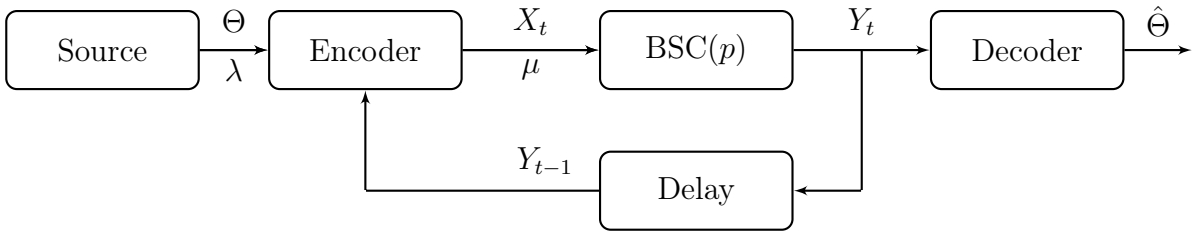

Figure 4.1: System model for causal encoding over the BSC with noiseless feedback. In "causal encoding" the source delivers information bits to the encoder as the transmission progresses, at a rate of λ [bits/sec], and the encoder sends coded bits over the channel at a rate of μ [bits/sec]. The entire information sequence is only available after the time taken by $K\frac{\mu}{\lambda}$ channel symbols has elapsed.

4.3 Contribution

The main contributions of this chapter are:

- A detail analysis of the regions where a causal encoder is needed to obtain an expected decoding time significantly lower than that of an of-the-shelf decoder not specifically designed as a causal encoder.

- An analysis of the performance of simple schemes or simple modification of non-causal algorithms, like bit repetition or independent segment transmissions, applied to the causal encoding problem. This analysis sets a benchmarks to compare the performance of a causal encoding algorithm.

- A causal encoder, the sub-block combining algorithm, that outperforms the expected decoding time of non-causal algorithms, and exhibits low run-time complexity and low memory complexity. The improved performance is achieved by combining partially decoded sub-systems into a single system, and the low runtime complexity is achieved by consolidating many low probability groups when new sub blocks are added to the system. Also, the memory usage is optimized by avoiding multiple copies of the same system, and using memory references instead.

4.4 Organization

The chapter is organized as follows: Sec. 4.5 formulates the causal encoding problem as a constraint integer optimization problem; Sec. 4.6 analyzes performance bounds and defines the regions where causal encoding could enable a lower expected decoding time by making a more efficient use of the transmitted symbols; Sec. 4.7 analyzes performance bounds for simple modifications of non-causal algorithms when applied to the causal encoding problem, and highlights the gap between the performance of these algorithms and the bounds of Sec. 4.6, that could be bridged by the proposed causal encoder; Sec. 4.8 details the properties a causal encoder needs to satisfy to lower the expected decoding delay beyond that of Sec. 4.7, and with a complexity feasible for practical implementation; Sec. 4.9 introduces the sub-block combining algorithm that exhibits the properties described in Sec. 4.8; Sec. 4.10 analyzes the sizes of the sub-blocks that optimizes the performance of the sub-block combining algorithm; Sec. 4.11 provides simulation results of the sub-block combining algorithm and Sec. 4.12 concludes the chapter.

4.5 Causal Encoding Problem Statement

The causal encoding problem consists of designing an encoder-decoder pair that satisfy a reliability constraint and a "causality" constraint, and produce and estimate of the transmitted message at the earliest expected decoding time. Let the decoding time, in channel symbols, be T_d and let ϵ be a small upper bound on the frame error rate. Denote by θ_1^j the first j symbols of the information sequence: $\theta_1^j = \theta_1, \theta_2, \dot{,}\theta_j$. Then, the "causal encoding" problem

can be formulated by:

$$\begin{aligned} &\text{minimize:} && \mathsf{E}[T_d] && (4.1)\\ &\text{subject to} && \text{reliability constraint:} && \Pr(\hat{\theta}\neq\theta) \leq \epsilon\,, && (4.2)\\ &&& \text{“causality” constraint:} && X_{t+1} = \mathcal{F}\big(\theta_0^{\lfloor t\frac{\lambda}{\mu}\rfloor}, y_1^t\big)\,. && (4.3) \end{aligned}$$

This dissertation proposes a sub-block combining algorithm that implements a “causal encoder” and that extends the systematic approach used for the sequential transmission case in Sec. 2 and Sec. 3. The sub-block combining algorithm is an efficient causal encoder that decodes with a lower expected decoding time $\mathsf{E}[T_d]$, compared to non-causal, of-the-shelf encoders, with a complexity low enough for to allow for practical implementation and simulation of blocklenghts up to a few thousand bits.

4.6 Model-Inherent Performance Bounds and Regions of Interest

The streaming information source and the noisy channel in the causal encoding model impose inherent bounds on the achievable decoding times. The time that the source takes to produce all K bits of the information sequence θ, given by $K\frac{1}{\lambda}$ seconds, or $K\frac{\mu}{\lambda}$ channel symbols, sets a lower bound on the decoding time T_d. The number of channel symbols needed to transmit a K-bit message with a transmission rate equal to the channel's capacity sets a bound on the expected decoding time $\mathsf{E}[T_d]$, given by $\frac{K}{C}$. Shannon [Sha48] shows that reliable communication at a rate higher than the channel capacity C cannot achieved. Thus, the expected decoding time must be above both bounds:

$$\mathsf{E}[T_d] \geq \max\big\{K\frac{\mu}{\lambda}, \frac{K}{C}\big\}\,, \tag{4.4}$$

These two bounds define the lower boundary of the achievability region, shown by the shaded region in Fig. 4.2.

A traditional, non-causal encoder, starts the transmission after the entire K-bit information sequence becomes available. The expected decoding time, in channel symbol times, of a traditional system is no lower than the sum of the first two bounds, the time $K\frac{\mu}{\lambda}$ the source takes to produce the entire information sequence and the $\frac{K}{C}$ symbols channel symbols that a capacity achieving scheme would take to communicate the information sequence across the channel. This bound is shown in Fig. 4.2 with the light blue dashed line labeled $K\frac{\mu}{\lambda} + \frac{K}{C}$.

A systematic algorithm, such as the systematic posterior matching described in Sec. 3, leads to a straightforward causal encoder, where the K systematic bits are transmitted as they become available and non-systematic bits are only transmitted after the entire information sequence becomes available to the encoder. This type of algorithm could be described as a systematically causal encoder (SCE). Causal encoding permits transmission of $K\frac{\mu}{\lambda}$ symbols during the time the traditional system is waiting for the K message bits to become available. When $\mu \geq \lambda$, The expected decoding time T_{SCE} of a SCE is lower bounded by the sum of the time needed for all K bits of the information sequence to arrive at the encoder and be transmitted, and the time needed by a capacity achieving scheme to transmit the $\frac{K}{C} - K$ non-systematic symbols. Let $\gamma \triangleq \frac{\lambda}{\mu}$ be the "normalized" symbol rate, then, the lower bound on the expected decoding time of an SCE is compactly described by:

$$\mathsf{E}[T_d] \geq \max\{K\gamma, K\} + \left(\frac{K}{C} - K\right). \tag{4.5}$$

The bound is comprised of two terms: the time until all K systematic bits have been transmitted (left term) and the time required to transmit the subsequent non-systematic bits (right term). The left term is characterized by two regions: one region where $\lambda \geq \mu$, or $\gamma > 1$ and another region in which $\lambda < \mu$ or $\gamma < 1$. Note that when $\lambda \geq \mu$, or $\gamma \geq 1$, the first term becomes just K, and the overall bound becomes the bound set by a standard capacity achieving scheme, given by $\frac{K}{C}$. Thus, a systematic algorithm like the one in Sec. 3 could perform as well as any causal encoder, with an expecting decoding time $\mathsf{E}[T_d]$ that

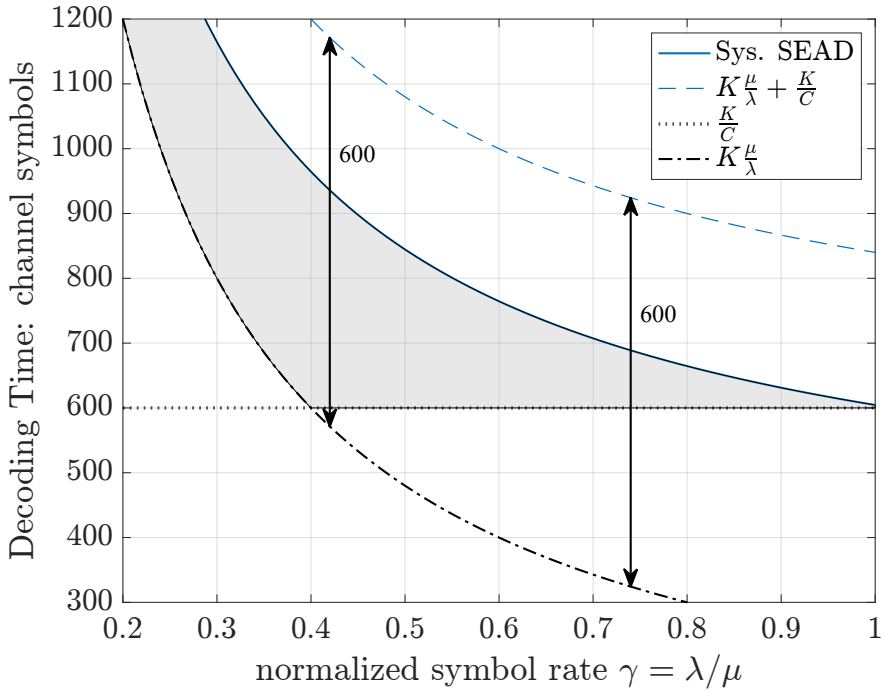

Figure 4.2: Shows the region where a "causal encoder" may achieve a decoding latency lower than that achievable with a standard decoder, see shaded region. The vertical axis shows decoding time, in transmitter symbols, and the horizontal axis shows the normalized symbol rate, defined by $\gamma = \frac{\lambda}{\mu}$. The regions described are for a channel with capacity $C = 0.40$, and for a message size $K = 240$. The shaded region is bounded above by the average decoding time from simulations of the systematic encoder that enforces the SEAD constraints, implemented in chapter 2, see the solid curve labeled "Sys. SEAD." And is bounded below by the the time, in transmitter symbols, that the source takes to deliver the entire information sequence, see the black dash dot $-\cdot$ curve defined by the expression $K\frac{\mu}{\lambda}$ in the label, and the expected decoding time performance of a system that both operates at capacity and starts the transmission as soon as the first information bit arrives, see the horizontal dotted line defined by $\frac{K}{C}$ in the label. The dash blue curve is the decoding time of a standard encoder that operates at capacity, but that starts the transmission after the entire information sequence arrives at the encoder, see the vertical lines labeled 600 with origin at the time where the last information bit arrives at the encoder.

approaches bound 4.5 as K becomes large and $\mathsf{E}[\tau] \to \frac{K}{C}$. For $\lambda \geq \mu$ the encoder of 3 may be applied directly and a "new" encoder is not needed.

However, if $\lambda < \mu$, or $\gamma < 1$, then $\frac{K}{\gamma} > K$ symbols (bits), may be transmitted by the time the full message becomes available, which is $\frac{K}{\gamma} - K$ non-systematic bits in addition to

the K systematic bits. This dissertation explores causal encoding to attempt to use all (or most) of the $\frac{K}{\gamma}$ available symbols to reduce the expected decoding time $\mathsf{E}[T_d]$.

This dissertation seeks to attain a decoding time below that achievable by an SCE, and thus the lower bound (4.5) defines the upper boundary of the region of interest, shown by the shaded region of Fig. 4.2. The systematic encoder described in Sec. 3 is an SCE encoder that approaches the BSC capacity rapidly as K becomes large, and thus approaches the performance of bound (4.5).

Conversely, when the transmission speed μ is much larger than the source speed λ, the problem becomes non-interesting. In this region, even repetition coding could work well, because it would allow to drive the bit error rate (BER) low enough to achieve the target frame error rate (FER). Let the BER be P_b, with repetition coding the FER becomes $\Pr(\theta \hat{=} \theta) = (1 - P_b)^K$, then:

$$P_b \leq 1 - (1 - \epsilon)^{1/K} \implies (1 - P_b)^K \geq 1 - \epsilon \tag{4.6}$$

The same martingale analysis of Sec. 3, without the communication phase fall back, can be used to find the number of bit repetitions τ_{bit} required to attain the BER P_b of (4.6) and the ratio $\frac{\mu}{\lambda}$ that achieves it:

$$\frac{\mu}{\lambda} \geq \tau_{bit} \triangleq \frac{\left\lceil \log_2 \left(\frac{(1-\epsilon)^{1/K}}{1-(1-\epsilon)^{1/K}} \right) \right\rceil}{C_2} \implies P_b \leq 1 - (1 - \epsilon)^{1/K} . \tag{4.7}$$

The focus of this chapter is the interesting region where a thoughtful approach is necessary. This region is $\lambda < \mu < \lambda\tau_{bit}$, where μ is larger than λ, but not large enough to use repetition coding. Also note that as the ratio $\frac{\mu}{\gamma}$ increases the BER gets smaller. In the limit, all but the last bit could be made arbitrarily reliable, and decoding could end when the error probability of the last bit is lower than ϵ. This case defines a tighter lower bound on the

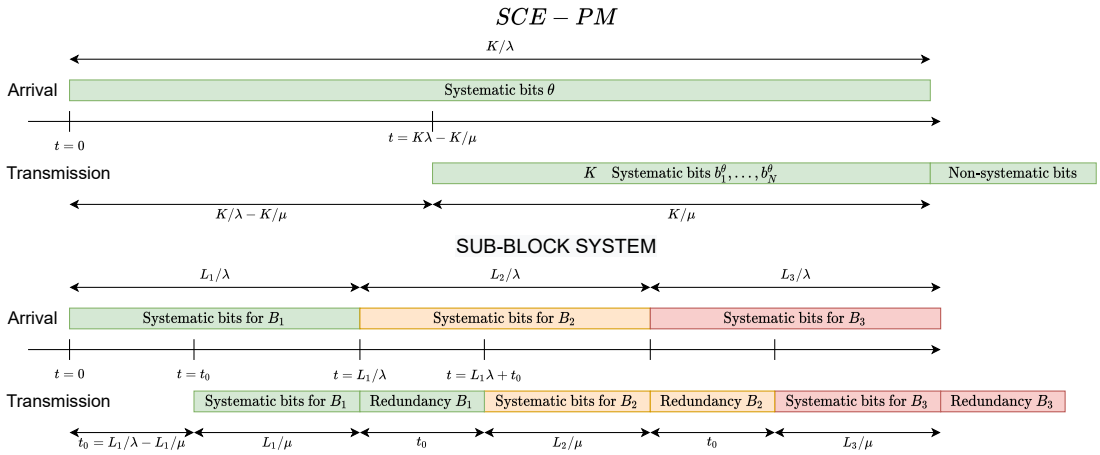

Figure 4.3: Shows the times of source symbol arrivals and the time that coded symbols may be transmitted, for a systematically causal encoder and a causal encoder that transmits non-systematic symbols while a subsequent block of source symbols become available. The figure is for a case where the transmitter rate μ is higher than the source rate λ, and thus a "causal" approach is relevant.

minimum number of symbols required to achieve a FER $\Pr(\theta \neq \hat{\theta}) \leq \epsilon$, and is given by:

$$\mathsf{E}[\tau] \geq \tau_l \triangleq \frac{K\mu}{\lambda} + \frac{\left\lceil \log_2\left(\frac{1-\epsilon}{\epsilon}\right)\right\rceil}{C_2} = \frac{K\mu}{\lambda} + \tau_{last}, \qquad \tau_{last} \triangleq \frac{\left\lceil \log_2\left(\frac{1-\epsilon}{\epsilon}\right)\right\rceil}{C_2} \tag{4.8}$$

An alternative to bit repetition that does not require a new encoding scheme is to break the K message bits into segments, and transmit each segment as a standalone message.

4.7 Independent Sub-blocks

Using a systematically causal encoder like the one in Sec. 3 to transmit the K-bit message into independent segments is a natural approach to the causal encoding problem. This approach allows to use some of the available symbols, beyond just the systematic bits. However, it has significant disadvantages. Suppose that N such segments are used, each of length L_j, $j = 1, 2, \ldots, N$, and an independent system B_i, $i = 1, 2, \ldots, N$ is used to decode each segment. The rate performance for this approach will be that of the system in Sec. 3 at the

block sizes L_i and for a higher reliability threshold. The smaller blocklegths lead to a rate loss. Note that, by linearity of expectations, the expected blocklenght will be the sum of the expectations at each segment, which is higher than the expected blocklength of a single system size K. The higher reliability threshold on each segment also leads to further rate loss, and is needed to achieve the error rate target $\Pr(\hat{\theta} \neq \theta) \leq \epsilon$ at the combined system. To see this, let $k_j \triangleq \sum_{n=1}^{j} L_j$, and let $\hat{\theta}^{(j)} \triangleq \hat{\theta}_{k_{j-1}}, \hat{\theta}_{k_{j-1}+1}, \ldots, \hat{\theta}_{k_j}$. For independent segments the probability $\Pr(\hat{\theta} = \theta)$ is given by the product of the probabilities $\Pr(\theta^{(j)} = \hat{\theta}^{(j)})$. If we use the same target for each segment then:

$$\Pr(\theta^{(j)} = \hat{\theta}^{(j)}) \geq (1-\epsilon)^{1/N} \qquad \Longrightarrow \qquad \prod_{j=1}^{N} \Pr(\theta^{(j)} = \hat{\theta}^{(j)}) \geq 1 - \epsilon \,. \tag{4.9}$$

The independent sub-blocks approach does improve the performance compared to repetition coding, by lowering the number of "confirmation phases" from the message size K to the number of segments N. An expression for the expected time needed for each confirmation phase, under a gene aided decoder, may be lower bounded with the methods of chapter 3, using a step size of C_1 instead of C, from the time T_0 when the correct message enters the confirmation phase for the first time. The expression for this lower bound is:

$$\mathsf{E}[\tau - T_0] \geq \frac{\log_2\left(\frac{(1-\epsilon)^{1/N}}{1-(1-\epsilon)^{1/N}}\right)}{C_1} \,. \tag{4.10}$$

The total decoding time increases, over the single block case, by a minimum of:

$$\log_2\left(\frac{(1-\epsilon)^{1/N}}{1-(1-\epsilon)^{1/N}}\right)\frac{N}{C_1} - \frac{\log_2\left(\frac{1-\epsilon}{\epsilon}\right)}{C_1} > \frac{N-1}{C_1}\log_2\left(\frac{(1-\epsilon)^{1/N}}{1-(1-\epsilon)^{1/N}}\right) \,. \tag{4.11}$$

To avoid the higher decoding times incurred by independent sub-blocks in the confirmation phase, this chapter studies "causal encoders" that combine separate segments into a single system before each segments enters the confirmation phase. An encoder that achieves this goals may avoid the $N-1$ additional "confirmation phases" and the increased length

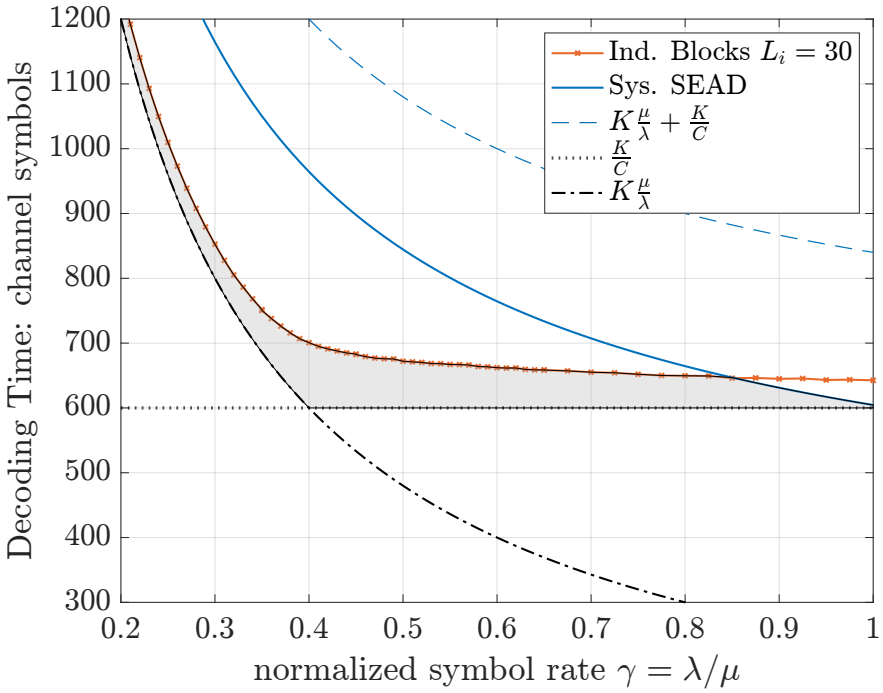

Figure 4.4: Performance of Independent Blocks. The new orange line $-\times$ added to the bounds of Fig. 4.2 shows the decoding time achievable via independent systematic posterior matching blocks. All 8 blocks use were the same size 30-bits, and the message size K is 240-bits. To the left, smaller blocks, and even bit repetition could perform better. To the right, The single block systematic posterior matching, blue solid line, performs better. A purpose built "causal encoder" can perform better lies only within the shaded region.

"penalty" associated with the higher "reliability" target $(1-\epsilon)^{1/N}$ imposed on each independent segment to achieve a combined reliability of $1-\epsilon$.

4.8 Requirements of a Causal Encoder

The analysis of bit repetition and independent sub-blocks allows to determine the properties that are desired in a "causal encoder" design. These properties include ability to encode channel bits, beyond just the systematic bits, without the full knowledge of the information sequence. The "causal encoder" must also be able to add new information (segments) to the process, in order to optimize future transmissions and avoid the penalty shown in (4.11). Lastly, the complexity of the "causal encoder" needs to be sufficiently low to make the

implementation feasible. Note that for the sequential case, low complexity was achieved by grouping messages that shared a posterior probability. However, this approach becomes problematic when two segments are combined into a larger system, which is exactly what the causal encoder must do. This is because the number of different posteriors becomes too large as new segments are combined. The challenges that combining segments pose are explained in detail next.

Suppose that two separate message segments are encoded by two posterior matching systems independently. It is possible combine the two systems into a single larger system. Let an item i of the larger system be comprised of segment $i^{(1)}$ in the first system and segment $i^{(2)}$ in the second. More generally, suppose candidate message i is comprised of N segments $i^{(1)}, i^{(2)}, \ldots, i^{(N)}$, each of length L_j, $j = 1, 2, \ldots, N$, and let $k_j \triangleq \sum_{n=1}^{j} L_j$ as before, with $k_0 = 0$, $K_1 = L_1$, and $k_N = K$. Let $Y^{j,t}$ be the symbols of Y^t received for posterior matching system j, and suppose they were ordered so that $Y^t = Y^{1,t}, Y^{2,t}, \ldots, Y^{N,t}$. The posterior of the entire $i = i^{(1)}, i^{(2)}, \ldots, i^{(N)}$ is given by:

$$\Pr(\theta = i \mid Y^t) = \Pr(\theta_{k_0}^{k_1}, \theta_{k_1}^{k_2}, \ldots, \theta_{k_{N-1}}^{k_N} = i^{(1)}, i^{(2)}, \ldots, i^{(N)} \mid Y^t) \tag{4.12}$$

$$= \prod_{j=1}^{N} \Pr(\theta_{k_{j-1}}^{k_j} = i^{(j)} \mid \theta_{k_j}^{k_{j+1}}, \theta_{k_{j+1}}^{k_{j+2}}, \theta_{k_N}^{k_{-1}}, Y^t) \tag{4.13}$$

$$= \prod_{j=1}^{N} \Pr(\theta_{k_{j-1}}^{k_j} = i^{(j)} \mid Y^{j,t}) \tag{4.14}$$

$$\rho_{j,i}(y^t) \triangleq \Pr(\theta_{k_{j-1}}^{k_j} = i^{(j)} \mid Y^{j,t}) \,. \tag{4.15}$$

Where (4.14) follows since $\{\theta_{k_{j-1}}^{k_j} = i^{(j)}\}$ is independent of $\{\theta_{k_{l-1}}^{k_l} = i^{(j)}\}$ and of $\{Y^{l,t} = y^{l,t}\}$ if $j \neq l$, since each $Y^{j,t}$ depends on the channel and $X^{j,t}$, and $X^{j,t}$ is a function of only $\theta_{k_{j-1}}^{k_j}$ and the segment feedback that is part of $Y^{j,t}$. Equation (4.14) describes how to compute the posterior of a message (or a message segment) that is comprise of independent segments with known posteriors. Extending this to a group of messages with shared posterior is straightforward, since it only requires to replace each $i^{(j)}$ with a group of n_j items $\mathcal{G}^{(j)} =$

$i_1^{(j)}, i_2^{(j)}, \ldots, i_{n_j}^{(j)}$ and i with a product set $\mathcal{G}^{(1)} \times \mathcal{G}^{(2)} \times \cdots \times \mathcal{G}^{(N)} = \{i_1^{(1)}, i_2^{(1)}, \ldots, i_{n_1}^{(1)}\} \times \{i_1^{(2)}, i_2^{(2)}, \ldots, i_{n_2}^{(1)}\} \times \cdots \times \{i_1^{(N)}, i_2^{(N)}, \ldots, i_{n_N}^{(1)}\}$ instead in the segment. The number of items that share the posterior $\prod_{j=1}^{N} \rho_{j,i_1}(y^t)$ is given by $\prod_{j=1}^{N} n_j$, as each item in segment j can form a candidate with every item in any other segment l. The systematic posterior matching algorithm of Sec. 3 starts with exactly $K+1$ groups right after the systematic transmissions, and grows by about one each transmission. If the segment systems are systematic posterior matching systems, the number of groups in a larger system that combines $n-1$ segments is at least $\prod_{j=1}^{n-1}(L_j+1)$. Each additional segment n that is combined, increases the number of groups in the combined system by a factor of $l_n \geq L_n+1$. To avoid this excessive complexity, this dissertation proposes a sub-block combining algorithm that is explained next.

4.9 The Sub-Block Combining Algorithm

The "causal encoder" proposed in this dissertation is a "sub-block combining algorithm" (SBC) that exhibits all the desired properties of a good "causal encoder" described in Sec. 4.8. The sub-block combining algorithm is a PM algorithm that synthesizes a larger-blocklength that combines a new PM system with an existing system. The new PM system carries information of newly arrived message bits, where the new message bits could be as small as a single bit or as large as the remainder of the K-bit message. The algorithm starts the transmission with as soon as a small message segment is available to efficiently uses the opportunities to transmit channel bits. Existing and new message segments are combined into a single system to avoid the performance loss associated with additional, and longer, confirmation phases required for independent subsystems. Also, the sub-block combining algorithm consolidates several groups of messages with different posterior into a single larger group, where the larger groups represent single "posterior groups" with structures designed to easily extract them, individually, or as smaller groups, when their posteriors become larger, or when required to balance the partitions before each transmission. Consolidating many

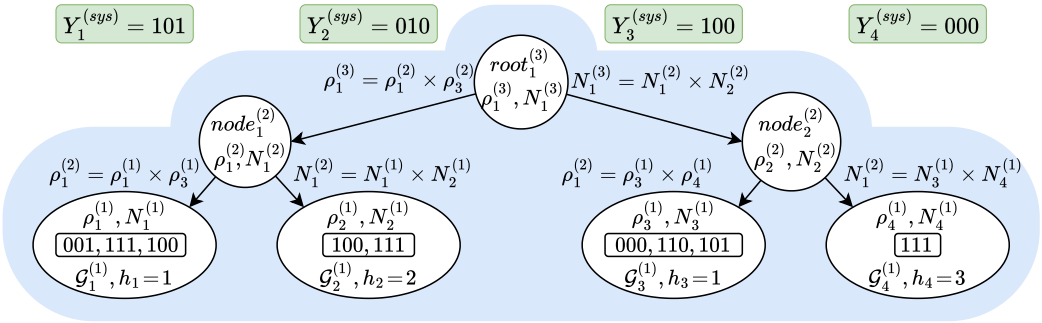

Figure 4.5: Shows a 2-Dimension slice of the tree structure used by the sub-block combining algorithm. The tree represents a set of $N_1^{(3)}$, where each message is comprised of segments from four sub-blocks with constant size $L_i = L_1 = 3$, and sharing the same posterior $\rho_i(y^t) = \rho_1^{(3)}$, see root node. The super indices denote the tree depth, from bottom to top, and the sub-indices denote the horizontal order that each node takes within the larger K-size message. The leaves are groups $\mathcal{G}_n^{(j)}$ of the basic SPM systems, with posteriors $\rho_n^{(1)}$, counts $N_n^{(1)}$ and Hamming distance h_n respect to the n-th systematic segment, see top $Y_n^{(sys)}$ segments. The figure shows how the posteriors $\rho_n^{(j+1)}$ and counts $N_n^{(j+1)}$, at higher level nodes $j = 2, 3$, are computed as a product of the values at the two children nodes, (or leaves) $\rho_n^{(j+1)} = \rho_{2n-1}^{(j)} \cdot \rho_{2n}^{(j)}$, and $N_n^{(j+1)} = N_{2n-1}^{(j)} \cdot N_{2n}^{(j)}$.

"posterior groups" into one allows the algorithm to avoid the high complexity associated with separately tracking every "posterior group" formed by messages that share a single posterior probability. The complexity of the sub-block combining algorithm is low enough to efficiently process messages with sizes up to a few thousand bits, as well as multiple configurations of number of sub-block and sub-block sizes.

The sub-block combining algorithm represents the messages with trees and lists of trees that combine multiple smaller message segment. For each tree, a child node could be a single tree, representing only messages segments with equal posterior like the tree shown in Fig. 4.5, or it could be a list of "sibling" nodes representing groups of messages with different posteriors like the trees in Fig. 4.7 and Fig. 4.8. The tree-list structure allows the sub-block combining algorithm to combine partially decoded systems into a single system, which may later be combined with another tree, to form an even larger system. This method allows the algorithm to maintain, under certain conditions, the computational complexity of the

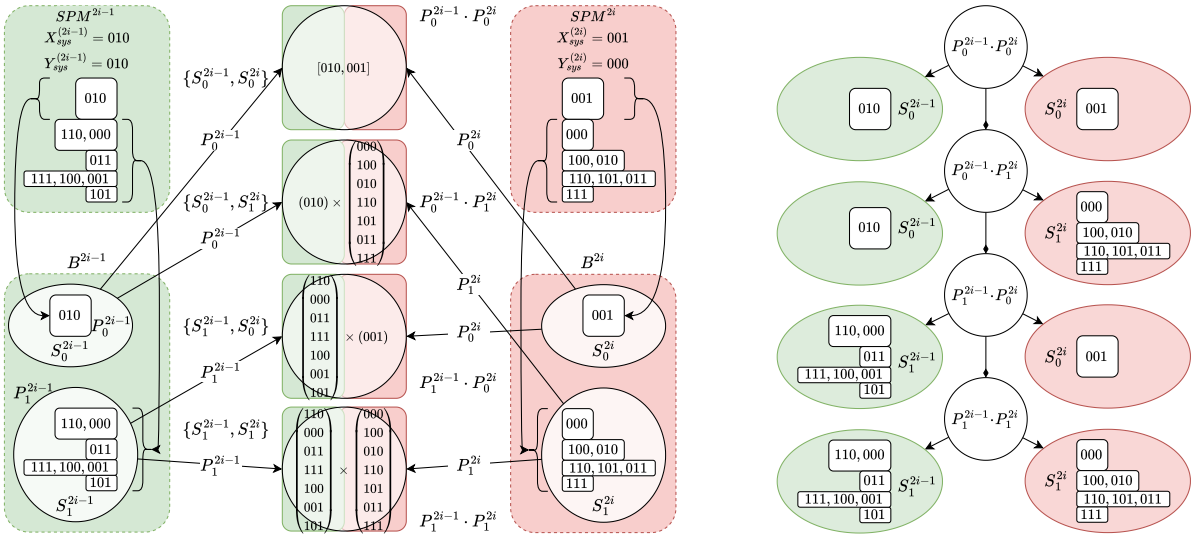

Figure 4.6: Construction of a list of four "trees" that combines two leaf sub-blocks systems with sub-block size of three, that are basic SPM systems, into a single system. The left side shows the components of each node and the right side shoes the final list of four "trees." First the leaf level lists, in the green and red top rectangles labeled SPM^{2i-1} and SPM^{2i}, are split into two sets, shown by the ellipsoids labeled S_0^{2i-1} and S_1^{2i-1} in the green rectangle labeled B^{2i-1} and the ellipsoids labeled S_0^{2i} and S_1^{2i} in the red rectangle labeled this B^{2i}. Note that the sets in the ellipsoids at each sub-block is used twice. This is because the message segments they contain form valid larger messages when combine with the components of each of the two segments on the other sub-block.

original PM system, in the sense of order $O(\cdot)$ as a function of blocklength.

Trees are natural choice to represent a message comprised of two or more segments. For example Schulman [Sch96] used trees in an interactive communication protocol. The sub-block combining algorithm uses binary trees because of the need to add a single new segment to an existing system. The "leaves" could be messages segments (or groups of messages segments) with shared posterior, not unlike the ones used in chapter 2. This makes the systematic posterior matching algorithm of chapter 2 a suitable building block for the "leaves" of the trees used by the sub-block combining algorithm. Each leaf j is a systematic posterior matching system operating on message segments j, from bit k_{j-1} to bit k_j, see the four leaves of the tree in Fig. 4.5 representing three bit message segments. The set of top

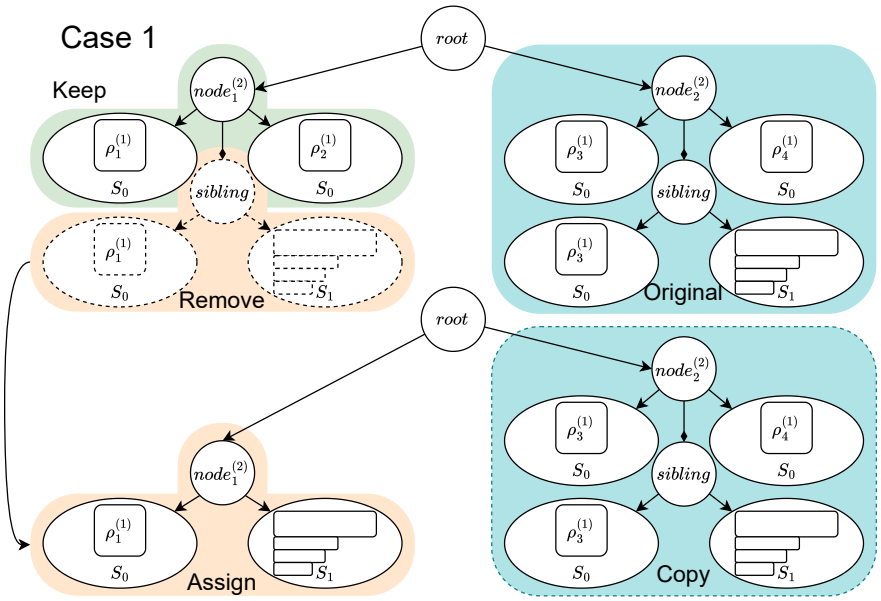

Figure 4.7: Shows how a tree of the sub-block combining algorithm is split into two smaller trees, when one or more node, that is not a leaf, has a "sibling" node. The top tree is the original tree and the bottom tree is the new tree that is created. The sepia colored "sibling" node with dashed lines on the left of the top tree is separated from the original tree and assigned as the left node of the new tree. Note tat the original tree has two "sibling" nodes on each side and each side. The messages represented by the top tree are defined by all four trees that can be constructed using a single node from each side. The two new trees together must represent all four trees, thus, "sibling" nodes of only one side can be extracted when splitting a tree into two, and the other side must remain the same for both trees. Thus, the right side of the bottom tree is the exact copy of the right side of the top tree.

level trees, that represent all the current message segments, are organized as a sorted list, see the list of trees with two "leaves" Fig. 4.6.

The tree architecture of Fig. 4.5 alone would still suffer from the complexity of a multiplicative number of trees. To address this problem, the sub-block combining algorithm allows each tree node to be itself a list of trees, with no restriction in the number of "siblings" or "links" in front of each node, see the "sibling" to the nodes the top tree in Fig. 4.7. Thus, each node could either be a single tree, or it could be the "head" node of the entire list of a posterior matching segment (or collection of segments). A node is constructed from two

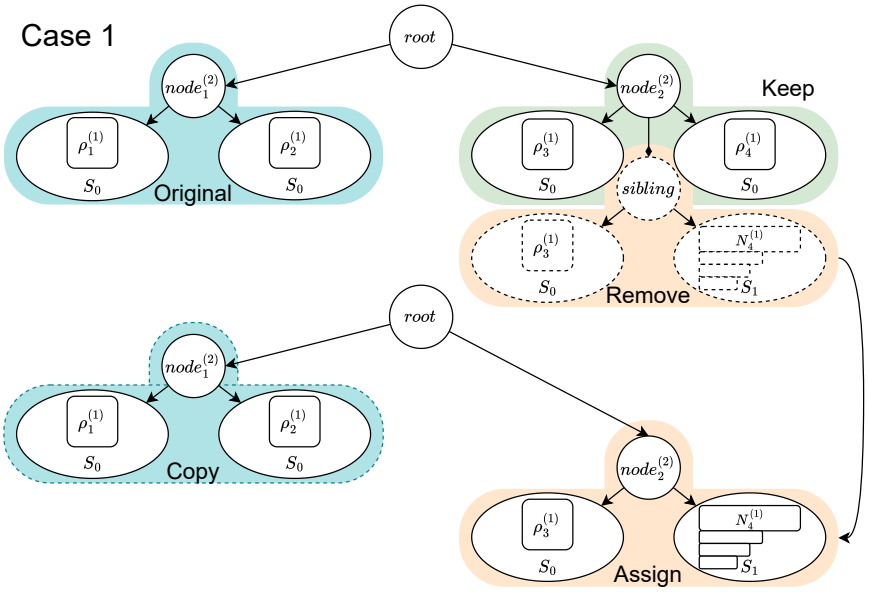

Figure 4.8: Shows how the sub-block combining algorithm splits into two trees a tree where a single non-leaf child node has a "sibling." The messages represented by the top tree are defined by the two trees that can be constructed using the single node at the left side and each "sibling" node from the right side, thus, the "sibling" node at the right side is extracted to form the right node of the new tree, while the left node of the new tree is an exact copy of the left node in the original tree. Note that the top tree is the top tree of Fig. 4.7 after the left sibling was extracted to construct a new tree.

systems, each with a list of nodes. First the two lists are split into two lists each, then, one tree is constructed for each of the four combinations of one list from the first system and one from the second system. This process starts at the leaf level posterior matching system, Fig. 4.6 shows how a larger system (right list of trees) is constructed from "leaf" systems $2i-1$ and $2i$. The list of "posterior groups" in systems SPM^{2i-1} (top left green rectangle) is split into S_0^{2i-1} and S_1^{2i-1} (the two ellipses in the bottom left green rectangle), and the list in system SPM^{2i} (top center red rectangle) is split into S_0^{2i} and S_1^{2i-1} (the two ellipses in the bottom center red rectangle). The four trees represents the messages combining $\{S_0^{2i-1}, S_0^{2i}\}$, $\{S_0^{2i-1}, S_1^{2i}\}$, $\{S_1^{2i-1}, S_0^{2i}\}$ and $\{S_1^{2i-1}, S_1^{2i}\}$, (see squares between the green and red rectangles), and the resulting list of trees with two leaves each is shown in the right side of Fig.

4.6.

The sub-block combining algorithm needs to allow the standard operations needed for posterior matching over the BSC as the algorithm of chapter 2. At each time t the set of messages needs to be partitioned into two sets, balanced according to some deterministic rule, before symbol X_t is encoded; the transmitter must identify which partitions contains the message θ to encode X_t; the posteriors need to be updated after symbol Y_t is received; and the receiver must be able to recover an estimate $\hat{\theta}$ at the end of the transmission. To build balanced partitions, the items in a node node that represents many messages may need to be allocated to a different partition, which requires splitting the node. These events will eventually arise as the accumulated posteriors of some nodes grow large. Maintaining the list of trees sorted in some order is desired to keep the complexity of partitioning low. The sub-block combining algorithm maintains the list of trees sorted according to the largest posterior in each tree. The two partitions also preserve this ordering, so that they may be combined again into a sorted list with low complexity after the posteriors are updated. The method to split a node, or tree, into two or more trees is explained next.

A tree representing multiple messages may need to be split into two parts, with a target posterior for one of the parts. This may be accomplished by splitting the tree into more than two trees, and making two smaller list of trees, where the total posterior in one of the two satisfies the target requirement, and the other collects the rest. Part of the posterior matching method consist of maintaining the posteriors for every message in the set Ω, which the sub-block-combining algorithm accomplishes via the the list of trees. Thus, the new trees obtained from splitting an original tree must be disjoint, and collectively represent exactly the same set of messages as the original top level tree. The same holds for every tree node. In a top level tree, any child node may be itself the head of a list of "siblings," in which case the tree represents items with different posteriors. To split these trees, the first time a node with siblings is encountered, the "head" node alone is kept in the original tree and the next node becomes the "head" node of a new tree. Every other branch, except the ones below the

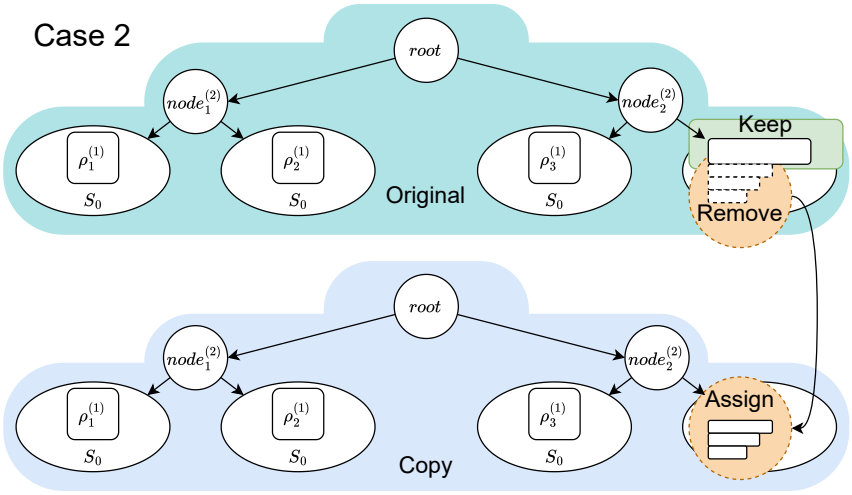

Figure 4.9: Shows how the sub-block combining algorithm splits into two trees a tree where only leaf node have "sibling" nodes. The collection of messages groups represented by the tree are defined by all the combinations that take one "sibling" leaf node from each leaf. The splitting process in this case is the same described in Fig. 4.7 and Fig. 4.8. Only the right leaf of the top tree has "sibling" leaves. To split it, all but the the first "sibling" are removed from the original tree, and assigned to the new tree, the bottom tree. All other leaves of the bottom tree are exact copies of those in the top tree. Note that the top tree is the top tree of Fig. 4.8 after the only remaining non-leaf "sibling" node has been removed and assigned to a new tree.

node at which the splitting happened, must remain the same for both new trees, so that the two new trees represent the same set of messages as the original top level tree. Examples of this type of tree splitting are shown in Fig. 4.7 and Fig. 4.8. Note that in Fig. 4.7 both the left and right "child" nodes have "siblings", but only one is split at a time. The example of Fig. 4.8 shows the case when only one "child" node has a "sibling" and this tree is exactly the first top tree that remained after splitting the tree of Fig. 4.7. The same process is used to split "leaf" level "siblings" after all no more parent node "siblings" exist in a tree, as shown in Fig. 4.10. Note that the new tree that keeps the original head "sibling" node will also maintain its original largest posterior value. In contrast, the tree that gets the next "sibling" node will have a new value for its largest posterior, which will be smaller than that of the original tree and will need to be inserted at the appropriate location in the list of trees, in

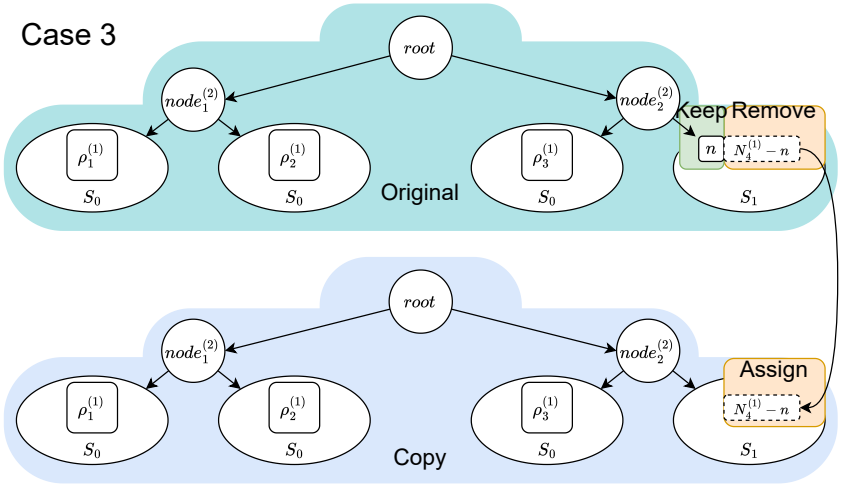

Figure 4.10: Shows how the sub-block combining algorithm splits into two trees a non-singleton tree where no node, including the leaf nodes, has a "sibling" nodes. All messages represented this type of tree, like the top tree with turquoise shade and the tree shown if Fig. 4.5, share the same posterior. The messages represented by the original tree, the top tree, are defined by all the combinations that take one segment from each leaf. In a non-singleton tree, at least one leaf needs to have a count of two or more messages. In the top tree, only the right leaf has a count $N_4^{(1)}$ greater than one. To split the tree, the algorithm computes the number n of message segments, where $1 \leq n < N_4^{(1)}$, that should remain on the original tree, and assigns all other segments to the new tree. All other leaves of the new tree, the bottom tree, are exact copies of those in the top tree. Note that the top tree is the top tree of Fig. 4.9 after all "sibling" leaf nodes have been removed and assigned to a new tree.

order to maintain the ordering.

The sub-block combining algorithm avoids the complexity and memory intensive operation of making copies of nodes with more than one "sibling" when splitting a tree with such nodes. To construct the new tree, only nodes with two or less "siblings" are copied. When the first node with more than two "sibling" is encountered, instead of copying the node and its "siblings," which may be a large list, the algorithm creates a single memory reference that points to the new head "sibling." No further operations are needed on the "children" of such nodes.

A with no remaining "sibling" nodes may still contain a "leaf" node representing a group

of more than one message segment. Every estimate represented by such tree will have the same posterior. To split this type of tree, a copy of the entire original tree is constructed, and one of the 'leaf" groups is split into two the same way as a group of SPM algorithm of chapter 2 is split. Since the count of items in the each group is known, the posterior of a tree is easier to tune to a target value, which is done by computing the number of "leaf" items to be assigned to each tree. This case is illustrated in the example of Fig. 4.10, where a single "leaf" contains a group with $N_3^{(1)}$ elements, and n items, where n is an integer such that $1 \leq n < N_3^{(1)}$, are kept in the original tree, while the remaining $N_3^{(1)} - n$ items are assigned to the new tree.

The number of top level trees in the list of trees may grow large, as the original trees are split into smaller trees before each transmission. However, when a new system is combined with an existing system to represent new message bit arrivals, the two lists are again consolidated into just four trees, and the process repeats. As the transmission progresses, trees whose total posterior grow larger pop to the top of the list, while rest of the trees fall to the bottom of the list, and are rarely operated on. This process affords the sub-block combining algorithm low enough complexity to efficiently combine many small segments to represent large messages.

The last step of the sub-block combining algorithm consists of reconstructed a single K-bits log message estimate $\hat{\theta}$ of the transmitted message θ. The estimate $\hat{\theta}$ constructed from a single tree that has single nodes, without "siblings" at every branch, including the "leaves," has a single item at each "leaf," and whose posterior has reached past the threshold $1 - \epsilon$. Such tree would be similar to the tree in Fig. 4.5, but with $N_i^{(1)} = 1 \quad i = 1, 2, 3, 4$, and where the product $\rho_1^{(1)} \cdot \rho_2^{(1)} \cdot \rho_3^{(1)} \cdot \rho_4^{(1)} \cdot \geq 1 - \epsilon$. Note that the right most "leaf" node in the tree of Fig. 4.5 represents a single 3-bit segment, while the others represent two or more 3-bit estimates. The algorithm recovers each segment of the estimate θ using the original SPM systems at each "leaf" level, that were described in chapter 2, and the indices of the current segment estimates from the selected tree. Then, all the estimate segments are combined to

form the larger K-bit estimate, which becomes the final decoding estimate θ.

4.10 Block Sizes

The sub-block combining algorithm does not require pre-defined sub-block sizes $L_1, \ldots, L_N$ at each "leaf" level segment. The algorithm may combine sub-blocks of any size and this property is exploited to select optimized sizes for the λ and μ values at which the system operates. The method to select appropriate sub-block sizes is described next.

Transmissions that only consider a segment of the message become inefficient when the segment is in its confirmation phase, according to the analysis of independent sub-blocks of Sec. 4.7. The confirmation phase of the current system, or just a "leaf" sub-block, starts when a posterior crosses $\frac{1}{2}$. The expected number of transmissions needed for "leaf" sub-block j of size L_j to produce a candidate $i^{(j)}$ with posterior $\rho_i^{(j)} \approx \frac{1}{2}$ is about $\frac{L_j}{C}$, according to (3.78) and the communication phase analysis of [YPA21]. Let γ be the normalize symbol rate $\gamma = \frac{\lambda}{\mu}$, the ratio of the source symbol rate and the transmitter symbol rate. The normalize symbol rate γ is the number of source bit arrivals during the time taken to transmit a channel symbol. The time needed for the L_j , $i = 1, \ldots, N$ systematic bits of each "leaf" sub-block B_j to arrive is $\frac{L_j}{\lambda}$, or $\frac{L_j}{\gamma}$ in channel symbols. The expected number of symbols needed for sub-block B_j to obtain a candidate l with $\rho_l^{(1)} \approx 0.5$ is about $\frac{L_j}{C}$, which takes $\frac{L_j}{C\mu}$ seconds, see bottom half of Fig. 4.3 labeled Sub-block System. From the $\frac{L_j}{C}$ symbols, L_j will be used for the systematic transmissions of "leaf" sub block B_j. The example of Fig. 4.3 shows the systematic transmission time $\frac{L_j}{\mu}$ of each sub-block in each the region where segments of the same color in the sub-block-system overlap. The remaining time $t_0 \triangleq \frac{L_j}{C\mu} - \frac{L_j}{\mu} = L_j \frac{1-C}{C\mu}$ is used for non-systematic transmissions. The transmission may become more efficient if the systematic transmissions of each next sub-block B_{j+1} start right after the first $\frac{L_j}{C}$ transmissions of block L_j. To attain the lowest possible decoding time T_s, the last systematic bit of the last sub-block B_N should be transmitted as soon as it arrives,

at time $t = \frac{K}{\lambda}$ or after $\frac{K}{\gamma}$ channel symbols. This criterion also applies to every sub-block B_j. The time used for the first $\frac{L_j}{C} - L_j$ non-systematic transmissions of sub-block B_j allows for $\frac{1-C}{C}\frac{\lambda}{\mu} = \gamma L_j \frac{1-C}{C}$ new source arrivals for the next sub-block B_{j+1}. The $L_{j+1}\frac{1}{\mu}$ seconds used for the L_{j+1} systematic transmissions of sub-block B_{j+1} allows for γL_{j+1} source bit arrivals. Then, at the start of the systematic transmissions of sub-block B_{j+1} there should be $L_{j+1}(1-\gamma)$ source bits already accumulated. The size L_{j+1} of the next sub-block, given the size L_j of the previous sub-block, needs to be tuned so that $L_{j+1}(1-\gamma)$ source bits arrive during the first $L_j\frac{1-C}{C}$ non systematic transmissions of sub-block B_j. Then, the each size L_{j+1} may be expressed in terms of the previous size L_j, the normalized symbol rate γ and the channel capacity C, via the following recursive equation:

$$L_{j+1}(1-\gamma) \leq L_j \frac{1-C}{C}\gamma \Longrightarrow L_{j+1} \leq L_j \frac{1-C}{C}\frac{\gamma}{1-\gamma}\,. \tag{4.16}$$

The recursive formula (4.16) defines an upper bound on each next sub block size L_{j+1} given the previous sub-block size L_j. The block sizes must be integer and must satisfy the constraint that their total is the message size K, that is:

$$\sum_{j=1}^{N} L_j = K\,. \tag{4.17}$$

Equations (4.16) and (4.17) define two constraints that the "leaf" sub-block sizes must satisfy and set a relation between the initial size L_1 (or final L_N), and the number N of "leaf" sub-blocks required:

$$K = \sum_{i=1}^{N} L_j \geq L_1 \sum_{i=1}^{N} \left(\tfrac{1-C}{C}\tfrac{\gamma}{1-\gamma}\right)^i = L_1 \frac{1-\left(\tfrac{1-C}{C}\tfrac{\gamma}{1-\gamma}\right)^{N+1}}{1-\left(\tfrac{1-C}{C}\tfrac{\gamma}{1-\gamma}\right)}\,. \tag{4.18}$$

When the factor $\frac{1-C}{C}\frac{\gamma}{1-\gamma} \approx 1$, which is when $\gamma \approx C$, constraint (4.16) dictates that the block sizes remain constant, note that the right most term of Eq. 4.18 is not well defined if $\gamma = C$.

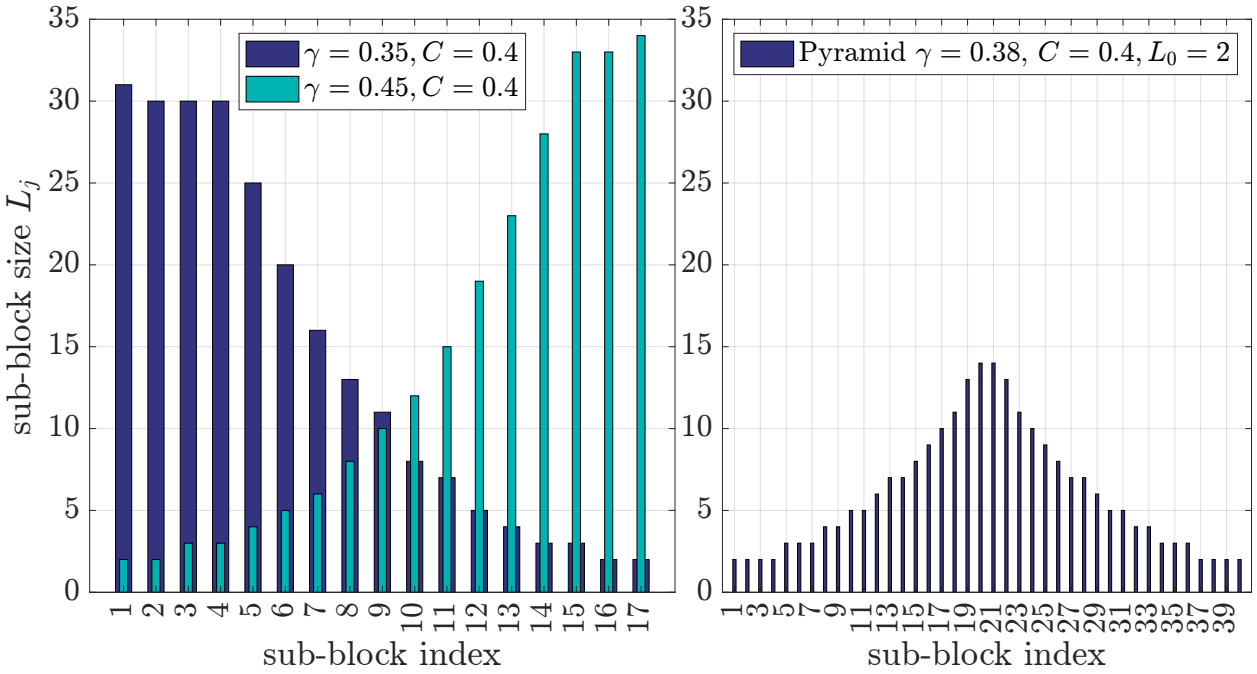

Figure 4.11: Sizes of the "Leaf" sub-block in the sub-block combining algorithm when the channel has capacity $C = 0.40$. Left: sizes that decrease, dark blue bars with height computed using Eq. 4.16 with $\gamma = 0.35$, and $L_N = 2$ designed for values of $\gamma \lesssim 0.35$; and sizes that increase, turquoise bars with height computed using Eq. 4.16 with $\gamma = 0.45$ and $L_1 = 2$, designed for values of $\gamma \gtrsim 0.45$. Right: sizes that increase and then decrease symmetrically to form a pyramid shape, computed using Eq. 4.16 with $\gamma = 0.38$ and $L_1 = 2$.

If the sizes of the "leaf" sub-block $L_j = L_1$, $\forall\ j = 1, 2, \ldots, N$ is constant, the number of "leaf" sub-blocks needed is just $N = \frac{K}{L_j}$. This case introduces an additional problem, small sub-block sizes leads to a large number N of sub-blocks and large sub-block sizes leads to a sub-optimal use of the starting and ending transmissions. An alternative choice when $\gamma \approx C$ is to select a desired number of "leaf" sub-blocks N and fixed small values for the initial and final sub-block sizes L_1 and L_N. For the other sub-block sizes $L_2, L_3, \ldots, L_{N-1}$ the block sizes may need to increase and then decrease. The sizes $L_2, L_3, \ldots, L_{N-1}$ may be computed using Eq. (4.16), for which appropriate values of γ for increasing and decreasing sizes must be selected to achieve the desired N. A natural choice for this case is to make the sizes symmetric about size $L_{\lceil N/2 \rceil}$.

Examples of block sizes are provided in Fig. and 4.11 for a message size $K = 240$ and

channel capacity $C = 0.4$. For the regions when $\gamma < C$ and $\gamma \approx C$, the initial block size was set at $L_1 = 2$. For the region when $\gamma > C$ the final block size was set at $L_N = 2$. The left of Fig. 4.11 shows the cases when $\gamma = 0.35 < C$ and $\gamma = 0.45 > C$. In both cases the number of blocks was 17. The right of Fig. 4.11 shows increasing and decreasing block sizes chosen for the case when $\gamma \approx C = 0.40$, where the initial and final block sizes were set at $L_1 = L_N = 2$. Then a value of $\gamma = 0.38 < C$ was used to compute the increasing block sizes and $\gamma = 0.42 > C$ to compute decreasing block sizes. The block sizes in Fig. 4.11 were used to obtain simulated performance results, and compared to constant block sizes $L_j = 1$ for reference.

4.11 Results

The sub-block combining algorithm was implemented to simulate the performance in expected decoding time $\mathsf{E}[T_s]$ for different normalize symbol rate γ values. The current implementation uses C++ and the simulations were executed on a MacBook Pro Laptop with an 8-Core Intel Core i9 2.4 GHz CPU. The performance results are illustrated Fig. 4.12, Fig. 4.13 and Fig. 4.14 The simulations show average decoding time $\mathsf{E}[T_s]$, in channel symbol times, vs. normalized symbol rate $\gamma = \frac{\lambda}{\mu}$. When $\mu = 1$ the number of channel symbols is also the time in seconds. Average decoding times for other values of μ may be obtained by simply multiplying the results in channel symbols by the symbol times $(\mu)^{-1}$. The simulations are over a channel with capacity $C = 0.40$ and for message size $K = 240$-bits.

The performance of the sub-block combining algorithm with the optimized choice of "leaf" sub-block sizes described in Sec. 4.10 and shown in Fig. 4.11 are shown in Fig. 4.12. For the region with $\gamma < C$, the choice "leaf" sub-block sizes decrease according to Eq. (4.16), and were computed using $\gamma = 0.35$. The performance of the algorithm on the region $\gamma < C$ is shown with the blue curve with $\circ$ markers, and the block sizes used are shown Fig. 4.11, dark blue bars with decreasing size on the left sub-plot. For the region

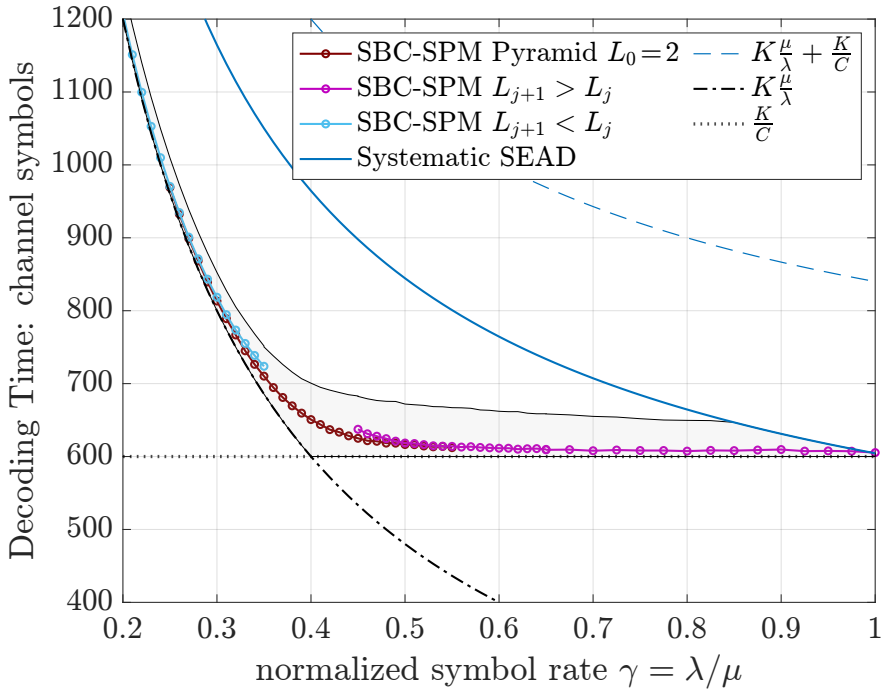

Figure 4.12: Performance of the sub-block combining algorithm with channel of capacity 0.4 and message length $K = 240$ compared to the region of interest from Fig. 4.4. The blue line $-\circ$ covering the region with $\gamma < C$ is for the decreasing block sizes of Fig. 4.11. The magenta line $-\circ$ covering the region $\gamma > C$ is for the block the increasing block sizes of Fig. 4.11. The burgundy line $-\circ$ covering the region where $\gamma \approx C$ is for the increasing and decreasing block sizes of Fig. 4.11 with $L_1 = L_N = 2$. .

with $\gamma > C$, the choice "leaf" sub-block sizes increase, also according to Eq. (4.16), and were computed using $\gamma = 0.45$. The performance on this region is shown with the magenta curve with $\circ$ markers, and the block sizes are the ones Fig. 4.11, turquoise bars with increasing size on the left sub-plot. The final block size is critical in the region where $\gamma \leq C$ and the initial block size is critical for the region where $\gamma \geq C$. Since constraint (4.16), sets upper bounds on L_{j+1} given L_j, the block sizes computed for $\gamma = 0.35 < C$ were used for the entire region $\gamma < 0.35$ and the block sizes computed for $\gamma = 0.45 > C$ were used for the entire region $\gamma > 0.45$. In the critical region $\gamma \approx C$, where $0.35 \leq \gamma \leq 0.45$, the performance of the sub-block combining algorithm is shown by the burgundy curve with $\circ$ markers. The block sizes on this region were allowed to increase and then decrease, where the increasing

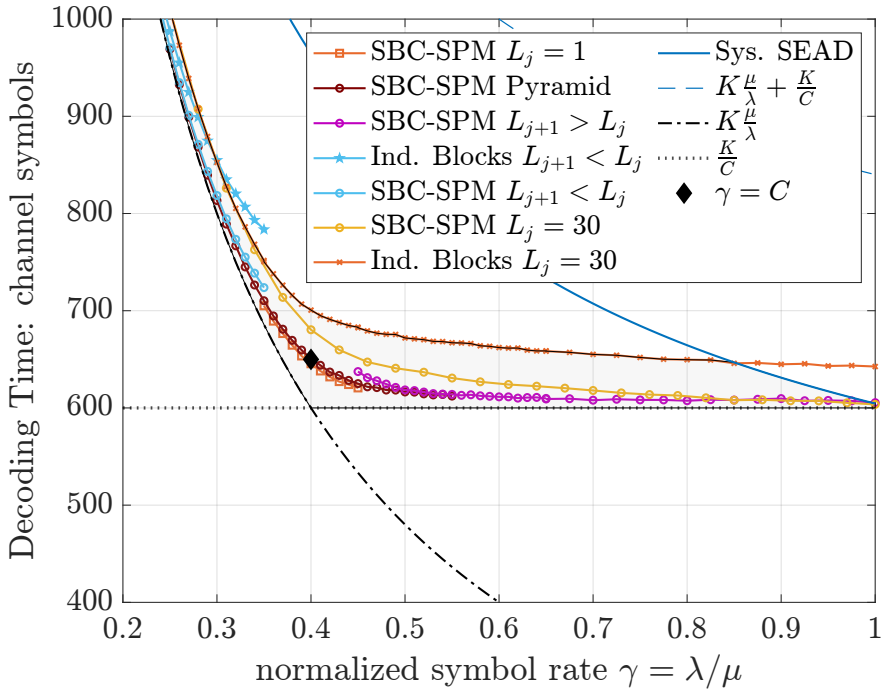

Figure 4.13: Performance comparison of the of the sub-block combining algorithm with optimized "leaf" block sizes and other choices of block sizes and algorithms. All curves are for a channel with capacity $C = 0.4$ and message length $K = 240$. In addition to the curves of Fig. 4.12, four curves are shown for reference. The yellow line with $\circ$ markers labeled SBC-SPM $L_j = 30$ shows the performance of the sub-block combining algorithm with 8-leaf sub-blocks of fixed size 30; the orange curve with $\square$ markers is for the optimal, but complex, choice of block size with 240 sub-blocks each size one; and the blue line with $\star$ markers labeled Ind. Blocks $L_{j+1} < L_j$ is the performance of decoding each block independently, for the choice of block sizes that was optimized for the sub-block combining algorithm. Finally, the black diamond highlights the point where $\gamma = C$.

sizes were computed using Eq. (4.16) with $\gamma = 0.38$ and $L_1 = 2$, and the decreasing sizes symmetrically reflect increasing sizes. Note that, over the region $\gamma \approx C$, changing block sizes do not conform to the constraint of Eq. (4.16), which dictates constant block sizes. The first two curves for $\gamma < C$ and $\gamma > C$ with only decreasing and only increasing sizes overlap with the curve for $\gamma \approx C$ to compare the performances. The sub-block combining algorithm with decreasing sizes closely approaches the bound set bu the source for values of $\gamma \ll C$. For $\gamma \gg C$, the algorithm also performs close to the bound set by the channel capacity, when increasing block sizes are used. Note that the algorithm with increasing and

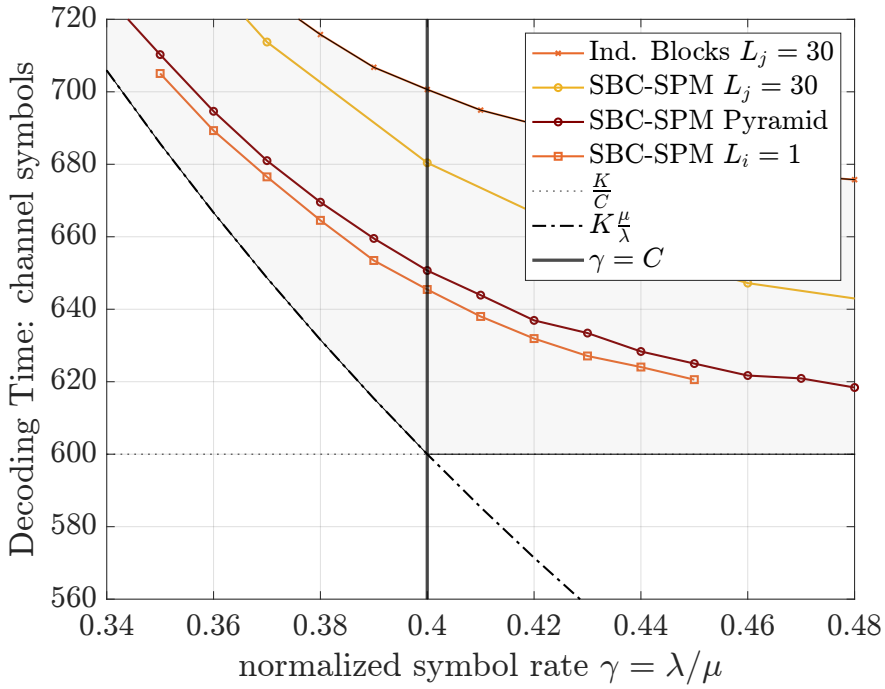

Figure 4.14: Detailed performance of the sub-block combining algorithm over the region where $\gamma \approx C$, also for a channel with capacity $C = 0.40$ and message size $K = 240$ bits. The Fig. show a zoom in of the same plot of Fig. 4.13, where the curves are described in detail. This Fig. highlights the significant improvements by the choice of block sizes that increase and decrease, shown in the right of Fig. 4.11, respect to fixed but large block size of 30, and the small gap between that choice and the optimal block sizes of one for all "leaf" sub-blocks.
.

decreasing block sizes outperforms only decreasing and only decreasing sizes in the overlap region, which may be explained by slower increase and decrease sizes computed with a value of γ closer to the channel capacity. The improved performance comes with an increase in number of blocks, and associated complexity increase.

The simulation results of Fig. 4.12 show that the sub-block combining algorithm performs very close the the bound set by the the source, in the region where $\gamma \ll C$ using decreasing block sizes. With increasing block sizes, the performance is very close to the bound set by the channel capacity in the region where $\gamma \gg C$. Block sizes that increase and then decrease size performs better over the region where $\gamma \approx C$, but not as close to the bounds set by the channel or the source. The block sizes in this region did not satisfy constraint (4.16).

To better evaluate and understand the performance of the sub-block combining algorithm, Fig. 4.13 provides three new curves, in addition to the independent sub-blocks curve of Fig. 4.4. The performance of the sub-block combining algorithm with the optimal size for $\gamma = C$, the smallest possible constant block sizes $L_j = 1,\ \forall j = 1, 2, \ldots, N$, where $N = K = 240$, is shown by the orange line $-\square$. This curve allows to evaluate the performance loss incurred by block sizes that increase and then decrease to reduce the number of "leaf" sub-blocks. Fig. 4.14 shows a close up of the region $\gamma \approx C$ to better compare these two curves. The curves show that the sub-block combining algorithm with increasing and decreasing block sizes performs very close to the optimal block sizes. The performance of the sub-block combining algorithm with 8 "leaf" sub-blocks of fixed size $L_j = 30,\ j = 1, 2, \ldots, 8$ is shown by the yellow curve with $\circ$ markers of Fig. 4.13. Note that this curve approaches the performance of increasing sub-block sizes when γ approaches 1, but increasing block sizes performs much better for smaller values of γ. To highlight the impact of combining the sub-blocks and optimizing the block sizes together, the performance of independent sub-blocks with the same increasing block sizes designed for the region with $\gamma < C$ is shown with the blue line $-\star$. This curve shows that independent sub-blocks performs well for very small values of γ, but the performance degrades rapidly as predicted by the analysis of 4.7, with a decoding delay even above that of sub-block combining algorithm with equal sub-block sizes $L_j = 30$, when $\gamma > 0.30$. The sub-block combining algorithm with sub-block sizes designed with $\gamma = 0.35$ performs well over most of the region where $\gamma < C$.

4.12 Conclusion

This chapter introduces the sub-block combining algorithm that implements causal encoding over the BSC with a sufficiently low complexity to allow for practical implementation. Causal encoding treats the communication scenario where transmission begins before all message bits are available. Message bits arrive at rate λ and codeword bits are transmitted at

rate μ. When $\lambda \geq \mu$, then systematic posterior matching can achieve the best possible performance. When $\lambda < \mu$, this chapter proposes a sub-block combining algorithm that provides an advantage over systematic posterior matching.

Breaking the message into sub-blocks enables efficient causal encoding without sacrificing the performance of the original systematic posterior matching algorithm. Both algorithms exhibit the same performance in the region $\gamma \geq 1$ where a "systematically causal" encoder suffices to efficiently used all the transmitted symbols, but the sub-block combining approach continues to provide excellent performance in the region where $\gamma < 1$, where the 'systematically causal" encoder suffers a significant loss in rate.

The sub-block combining algorithm elegantly groups messages with equal posteriors, as well as messages with different posteriors, into binary trees and lists of trees to keep the complexity manageable. This chapter also analyzes the sub-block sizes that optimize the decoding-time performance of the sub-block combining algorithm. The algorithm with sub-block sizes designed specifically for each region or γ exhibits improved expected decoding time as compared to algorithms with fixed sub-block sizes. The sub-block combining algorithm outperforms independent encoding of sub-blocks even with optimized sizes. Thus it is crucial to combine sub-blocks when their partitions include singletons. Lastly, the sub-block combining algorithm may be a suitable design for settings other than "causal encoding," and one such setting is the sparse feedback problem discussed in Ch. 5.

CHAPTER 5

Sparse Feedback Times

Consider the problem of communicating a K-bit message Θ through the BSC, where noiseless feedback is available on demand, instead of after every transmitted symbol. The sparse feedback times model is depicted in Fig. 5.1. At times $t = 1, 2, \ldots, \tau$. The encoder transmits coded bits X_t to the decoder through BSC at at symbol times The decoder receives binary symbols $Y_1, Y_2, \ldots$ that are noisy versions of X_1, X_2 where $\Pr(Y_t = 1 \mid X_t = 0) = \Pr(Y_t = 0 \mid X_t = 1) = p$. A noiseless feedback channel is available to the receiver to send the the symbols $Y_1, Y_2, \ldots$ back to the encoder where they may be used to encode future channel symbols as in chapters 2-4. However, in contrast to chapters 2-4, the receiver wishes to limit the number of times the feedback channel is used, and not the number of symbols transmitted each time. Thus, the receiver may allow a few symbols to accumulate, and later send them in packet

The decoder needs to produce an estimate $\hat{\Theta}$, of the transmitted message Θ, using the received symbols, and the process ends at a stopping time τ when the receiver is sufficiently confident of the estimate $\hat{\Theta}$. Let the total number of feedback packets be η, and let the times feedback packets are transmitted be $s_1, s_2, \ldots, s_\eta$. Let D_l be the number of bits in the packet transmitted at time $t = s_l$, where $s_0 = 0$, $s_\eta = \tau$ and for each $l = 1, \ldots, \tau$ $D_l = s_l - s_{l-1}$.

Let ϵ be a small threshold on the error probability, the sparse feedback times problem consists of producing the estimate $\hat{\Theta}$ at the receiver, with bounded error probability: $\Pr(\hat{\Theta} \neq \Theta) < \epsilon$ using the smallest possible average number of transmissions $\mathsf{E}[\tau]$ and average number of feedback packets $\mathsf{E}[\eta]$.

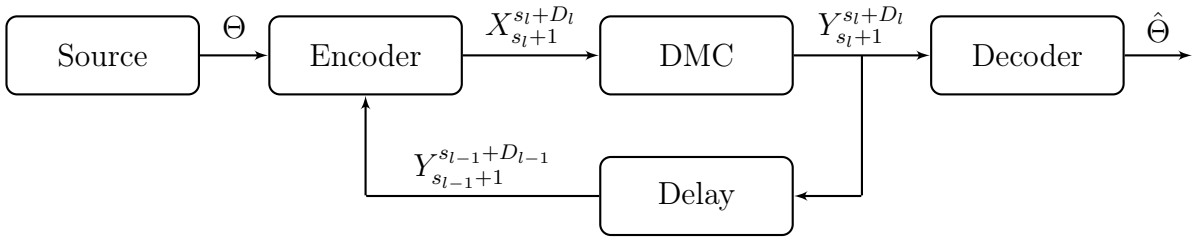

Figure 5.1: Sparse feedback times system model. The model is similar to the feedback communication models of chapters 2 and 3, but the feedback symbols may be held at the receiver for a few transmissions and then sent to the encoder in a packet Feedback transmissions happen at sparse times $t = s_1, s_2, \ldots, s_\eta = \tau$, and each feedback transmission, at a time s_l is a packet size D_l, containing the symbols $Y_{s_l+1}, Y_{s_l+2}, \ldots, Y_{s_l+D_l}$, where $s_{l+1} = s_l + D_l$.

5.1 Background

This chapter studies feedback communication with sparse feedback times for the same short blocklength regime of chapters 2-4 where variable length transmission with feedback is relevant. Practical communications systems that use feedback typically employ a hybrid ARQ architecture [Ric94], [LMS07]. An initial packet is sent, with subsequent packets of incremental redundancy sent only when needed, based on the feedback. In contrast, many theoretical studies consider systems with feedback *after every symbol*. Sparse feedback times have been studied within the context of stop feedback, where only an ACK or NACK symbol is used by the receiver to inform the transmitter when transmission should be terminated. In the sparse feedback scheme studied in this dissertation, the feedback is more rich than a simple ACK or NACK symbol, and communicates to the transmitter not only about when to stop but also about what encoding will be most informative.

Sparse stop feedback schemes have been studied in many works. Polyanskiy *et al.* [PPV10] introduced bounds for variable length stop feedback (VLSF) codes. Vakilinia introduced sequential differential optimization (SDO) for sparse stop feedback in [VRD16]. See also [YYK22] and [WWB18]. Yavas and Kostina [YKE24] developed achievability bounds for VLSF codes as a function of the error probability, average decoding times and number of

decoding attempts, for point-to-point, multiple access and random access channels. A bound for a similar scheme over the Gaussian channel was also introduced by Yavas *et al.* [YKE21].

The sparse, rich feedback setting was recently studied by Chen *et al.* [CYK23], as a bursty feedback scheme. As with SDO for stop feedback, in the setting studied by Chen *et al.* the feedback times are selected from a fixed and finite set, where the next feedback packet is only sent if it is needed to achieve reliable decoding. Chen *et al.* analyzed achievable expected rate as a function of the error probability and the maximum number of feedback times, and showed that their achievability rate bounds are higher than the bounds for stop feedback by Polyanskiy *et al.* [PPV10] when a maximum of 3-5 feedback packets are allowed. That is, a few sparse feedback times of rich feedback can facilitate higher rates that stop feedback after every symbol. For example, in the very low error probability regime, of about 10^{-10}, Chen *et al.* showed that sparse feedback with 5 rich feedback packets are sufficient to outperform unlimited stop feedback over blocklengths of up to 400 bits. In contrast, this chapter studies a potentially unlimited number of sparse feedback times for "posterior matching" communication. Stochastic (rather than fixed) feedback times are selected to guarantee an expected rate.

5.2 Contributions

This chapter proposes a version of "posterior matching" with stochastic feedback times, that lifts the restriction of classic posterior matching, where every channel symbol sent from the transmitter to the receiver is followed by a feedback symbol from the receiver to the transmitter. The proposed "posterior matching" scheme allows the receiver to wait for a few transmissions before sending the accumulated feedback symbols in a single packet The encoder must encodes the symbols transmitted between two feedback packets using only the feedback symbols received in previous feedback packets and these symbols may be transmitted in a single forward packet To make the proposed "posterior matching" scheme feasible,

this chapter introduces the "weighted median absolute difference" partitioning rule, a new set of encoding constraints that are less restrictive and better suited for block transmissions. The "weighted median absolute difference" rule guarantees a rate performance above one of the lower bounds analyzed non-sparse "posterior macthing" from Ch. 3, given by Eq. (3.27). This chapter provides the "look-ahead" implement sparse feedback times communication. To enable the transmission of a packet of symbols, the "look-ahead" algorithm encodes a few symbols in advance, which are guaranteed to satisfy the "weighted median absolute difference" encoding constraints. This chapter provides simulation results that show the the sparsity performance of the "look-ahead" algorithm with a rate performance indistinguishable from that of the systematic posterior matching algorithm of Ch. 2, which is above all the rate bounds of Ch. 3. This chapter provides a version of the "sub-block combining" algorithm adapted for sparse feedback communication. The complexity of the sparse "sub-block combining" algorithm is much lower than that of the "look-ahead" algorithm, and performs well at blocklengths of a few hundred bits, where the complexity of the "look-ahead" algorithm makes it prohibitively expensive.

5.3 Organization

The remainder of this chapter is organized as follows: Sec. 5.4 describes the communication approach by Naghshvar *et al.* in which the "look-ahead" algorithm is based. Sec. 5.5 formally introduces the sparse feedback times problem addressed in this dissertation. Sec. 5.6 introduces the "weighted median absolute difference" rule and the "weighted median" theorem. Sec. 5.7 introduces the "look-ahead" algorithm that implements sparse feedback times communication using the "weighted median absolute difference" rule. Sec. 5.8 proves the "weighted median" theorem that shows that the "weighted median absolute difference" rule guarantees the same performance as Thm. 1. Sec. 5.9 describes the sparse feedback times version of the "sub-block combining" algorithm of Ch. 4. Sec. 5.10 provides simulation

results for the "look-ahead" and sparse "sub-block combining" algorithms and Sec. 5.11 concludes the chapter.

5.4 Communication Scheme by Naghshvar *et al.*

The sparse feedback encoding algorithm proposed in this dissertation implement the "posterior matching" scheme proposed by Naghshvar *et al.* [NJW15] for general discrete memoryless channels. A brief description of the scheme follows. Both encoder and decoder use the channel symbol sequence up to the current time t: $\mathbf{Y}^t = Y_1, Y_2, \ldots, Y_t$ to compute the posterior probability $\rho_i(y^t) = P(\theta = i \mid Y^t = y^t)$, introduced in Eq. (2.3), and log likelihood ratio $U_i(t) = \log_2\left(\frac{\rho_i(Y^t)}{1-\rho_i(Y^t)}\right)$, introduced in Eq. (2.19), for each possible input message $i \in \Omega$. To encode the symbol X_{t+1} the encoder partitions the message space Ω into "bins", one for each possible input symbol, using a deterministic method known to the decoder. The encoder then transmits the symbol of the bin containing the transmitted message θ. The process terminates once a posterior crosses the threshold $1 - \epsilon$ and the message with this posterior is selected as the estimate. The choice of deterministic partitioning determines the scheme's performance and thus is at the core of the scheme. For the BSC, Naghshvar *et al.* [NJW15] proposed a partitioning algorithm that is described in chapter 2. For other channels Naghshvar *et al.* proposed the extrinsic Jensen-Shannon divergence as a metric to evaluate any set of bin constructions. The "look-ahead" algorithm proposed in this dissertation borrows from Naghshvar *et al.* the approach of using "bins" to encode a packet of symbols. The "look-ahead" algorithm consists of deterministic method to construct the "bins" that guarantees a rate performance, and this method is one of the main contributions of this chapter, along with the sparse version of the "sub-block combining" algorithm.

5.5 Sparse Feedback Times Problem

In the sparse feedback times model, the receiver may wait for a few symbols to accumulated before sending feedback back to the encoder. Then, the received symbols that accumulated between feedback transmissions are sent in a single packet The sparse feedbac times model is depicted in Fig. 5.1. The time between feedback transmissions could be variable, just like the block size. Let the feedback transmissions be at times $t = s_1, s_2, \ldots, s_\eta$, with $s_0 = 0$ and $s_\eta = \tau$. Then, at every time $t = s_{l+1}$ the receiver sends the feedback transmissions corresponding to times $s_l + 1, s_l + 2, \ldots, s_{l+1}$, in a block of size $D_l \triangleq s_{l+1} - s_l$, shown by the block $Y_{s_l+1}^{s_l+D_l}$ in Fig. 5.1.

The sparse feedback times communication problem consists of designing a variable length coding scheme to transmit a K-bit message using the smallest expected number of channel bits τ and the smallest expected number of feedback transmissions η that guarantees a frame error rate FER bounded by a small threshold ϵ. Note that the expectations $\mathsf{E}[\tau]$ and $\mathsf{E}[\eta]$ cannot be minimized at the same time. For instance, any forward error correction scheme that guarantees the FER bound ϵ achieves $\mathsf{E}[\eta] = 0$. However, as shown by Burnashev [Bur76] feedback and variable rate coding lower the error exponent, which achieves a target FER with a smaller $E[\tau]$. To formulate the communication problem the trade-off between $\mathsf{E}[\tau]$ and $\mathsf{E}[\eta]$ needs to be established.

There are many ways to formulate the problem and establish the trade-off between $\mathsf{E}[\tau]$ and $\mathsf{E}[\eta]$. A natural approach is using Lagrange multipliers, in which an expression of the form $\mathsf{E}[\tau] + \lambda\mathsf{E}[\eta]$ is minimized for some λ. The value of λ could represent the channel access cost, in transmission bits. However, even minimizing $\mathsf{E}[\tau]$ is an integer programming problem whose solution is not yet known. The approach to the sparse feedback times problem in this dissertation consists of designing a scheme that aims to minimize $\mathsf{E}[\eta]$ while attaining the expected block-length $\mathsf{E}[\tau]$ that satisfies the bound from (3.27). Suppose the bound on $\mathsf{E}[\tau]$

is τ_B, then the sparse feedback times problem can formulated follows:

$$\begin{aligned} &\text{minimize} && \mathsf{E}[\eta] && (5.1)\\ &\text{subject to:} && \mathsf{E}[\tau] \leq \tau_B,\ \Pr(\hat{\theta} \neq \theta) \leq \epsilon && (5.2)\\ &\text{Sparsity Constraint:} && \mathbf{X}_{s_l+1}^{s_{l+1}} = \mathcal{F}(\theta, \mathbf{y}_1^{s_l})\,. && (5.3) \end{aligned}$$

The sparsity constraint restricts the encoder to encode symbols $X_{s_l+2}, X_{s_l+3}, \ldots, X_{s_{l+1}}$ without using the feedback symbols $Y_{s_l+1}, Y_{s_l+2}, \ldots, Y_{s_{l+1}-1}$ not yet re-transmitted by the decoder. The challenge is to find an encoding function that guarantees that constraints (5.2) and (5.3) are satisfied and seeks to maximize sparsity in the feedback transmission times.

5.6 The "Weighted Median Absolute Difference" Rule

The next theorem relaxes the tolerance in the difference of sums (3.32), sufficient to guarantee constraints the rate the bound on $\mathsf{E}[\tau]$ in Eq. (3.27).

Theorem 6: The "Weighted Median" partitioning. *At each time t let S_0 and S_1 be a binary partition of Ω. Let $P_0 \triangleq \Pr(\theta \in S_0 \mid Y^t = y^t) = \sum_{i \in S_0} \rho_i(y^t)$, $P_1 \triangleq \Pr(\theta \in S_1 \mid Y^t = y^t) = \sum_{i \in S_1} \rho_i(y^t)$ and let $\Delta \triangleq \sum_{i \in S_0} \rho_i(y^t) - \sum_{i \in S_1} \rho_i(y^t)$, defined exactly as in chapter 2. Since $P_0 + P_1 = 1$, then $P_0 = \frac{1+\Delta}{2}$ and $P_1 = \frac{1-\Delta}{2}$. Let $\{o_1, \ldots, o_M\}$ be an ordering of the vector of posteriors such that $\rho_{o_1}(t) \geq \rho_{o_2}(t) \geq \cdots \geq \rho_{o_M}(t)$, and let m be the index of the "weighted median" posterior defined by:*

$$\sum_{i=1}^{m-1} \rho_{o_i}(y^t) < \frac{1}{2} \leq \sum_{i=1}^{m} \rho_{o_i}(y^t)\,. \tag{5.4}$$

Suppose that at every time t the "weighted median" absolute difference partitioning rule is

satisfied, along with the singleton rule (3.33) *given by* $\rho_i(y^t) \geq \frac{1}{2} \implies S_0 = \{i\}$ *or* $S_1 = \{i\}$*:*

$$\textit{"Weighted median" absolute difference rule: } \Delta^2 \leq \frac{2}{5}\rho_{o_m}(y^t) \tag{5.5}$$

Then, the constraints (3.22) (3.23) (3.24), (3.25) *and* (3.26) *of Thm. 1 are satisfied, and expected decoding time* $\mathsf{E}[\tau]$ *is upper bounded by bound* (3.27)*. The proof is in Sec. 5.8*

Rule (5.5) offers two significant advantages over SEAD and SED: the first is a larger tolerance on Δ, for most times s_l, since $\sqrt{\frac{2}{5}\rho_{o_m}(y^t)}$ is often much larger than $\rho_{o_m}(y^t)$. The second advantage is that the bound on Δ does not depend on which items are in S_0, which allows to allocate items to S_0 and S_1 to tune Δ without affecting the tolerance, unlike SED and SEAD in (2.9), (3.32) where changes in the partitioning cause changes in the tolerance.

5.7 The "Look-Ahead" Algorithm

The "look-ahead" algorithm is a method to design the partitions S_0 and S_1 for the next few transmissions s_l+1, $s_l+2, \ldots, s_l+D_l$ for some D_l, based only on the feedback symbols $Y_1^{s_l}$ received up to time $t = s_l$. The "look-ahead" algorithm needs to guarantee that constraint (5.5) is satisfied at each $t = s_l+1, s_l+2, \ldots, s_l+D_l$, for the already received sequence y^{s_l} and for each future possible extension sub-sequence $Y_{s_l+1}^{s_l+j}$, $j = 1, 2, \ldots, D_l - 1$. The key challenges that the "look-ahead" algorithm faces and the methods to overcome these challenges are described next.

First note that at any time $t = s_l$, only a few D_l values might be feasible, thus, a feasible value must be found before designing the partitions. Binary partitioning with $D_l = 1$ is always feasible as shown in chapter 3, and methods to construct binary partitions are shown in chapter 2, like the SED apgorithm by Naghshvar *et al.* [NJW15] or the thresholding of ordered posteriors. Second, the algorithm must always converge to a solution in a finite number of steps, preferably a small number. For this reason, a single attempt for a given

D_l will be executed, and upon failure, the value of D_l will be reduced by one before trying again. This procedure could fall back to the non-sparse two way partitioning where $D_l = 1$. Third, if S_0 and S_1 are fixed for the next times $t = s_l + 1, s_l + 2, \ldots s_l + D_l - 1$, then each future $\rho_{o_m}(y^t)$ and Δ is a random function of $Y_{s_l+1}, Y_{s_l+2}, \ldots, Y_{s_l+D_l-1}$, the future received symbols. The "look-ahead" algorithm needs to guarantee that the pair $\rho_{o_m}(y^t)$ and Δ satisfies constraint (5.5) at the current time s_l any any future time up to $s_l + D_l - 1$.

To overcome these challenges, the "look-ahead" algorithm proceeds as follows: let the 2^{D_l} "bins" at time $t = s_l$, be $\mathcal{E}_k$, $k = 0, 1, \ldots, 2^{D_l} - 1$ and define "bin" posteriors $P_{\mathcal{E}_k}$, δ_k, and $\delta_{\max}$ by:

$$P_{\mathcal{E}_k} \triangleq \sum_{i \in \mathcal{E}_k} \rho_i(y^{s_l}),\ \delta_k \triangleq P_{\mathcal{E}_k} - 2^{-D_l},\ \delta_{\max} \triangleq \max_k \{|\delta_k|\}, \tag{5.6}$$

where 2^{-D_l} is the target posterior for each bin. To overcome the uncertainty on $\rho_{o_m}(y^t)$ and guarantee that constraint (5.5) on Δ is satisfied at future time $t = s_l + 1, s_l + 2, \ldots, s_l + D_l - 1$ the algorithm finds a lower bound $\rho_{o_m}^{\min}(y^t)$ on $\rho_{o_m}(y^t)$ that is used to compute an upper bound $\Delta_{\max}$ on Δ for each future time up to $D_l - 1$. The method to find $\Delta_{\max}$ and a feasible D from an initial D is described next, and is also described in Algorithm 8. The "look-ahead" algorithm then uses $\Delta_{\max}$ to determine $\delta_{\max}$ the largest difference δ_k between the posterior $P_{\mathcal{E}_k}$ and the target 2^{-D_l}. Note that at each time $s_l + j$, $j = 0, 1, \ldots, D_l - 1$ each set S_x, $x \in \{0, 1\}$ collects the half of "bins" whose label has x at entry j, Then, Δ at time $s_l + j$ is given by:

$$|\Delta| = |\sum_{\mathcal{E}_k \in S_0} \delta_k - \sum_{\mathcal{E}_k \in S_1} \delta_k| \leq 2^{D_l} \delta_{\max} \tag{5.7}$$

Since $\rho_{o_m}(y^{t+1})$ depend on Δ at time t, an initial $\Delta'_{\max}$ is used to compute $\rho_{o_m}^{\min}(y^t)$, and then bounds $\Delta_{\max}$ on Δ and $\delta_{\max}$ on each δ_k, $t = s_l, s_l + 1, \ldots, s_l + D_l - 1$ are obtained via:

$$\Delta_{\max} \triangleq \min\{\Delta'_{\max}, \sqrt{\frac{2}{5} \rho_{o_m}^{\min}(y^t)}\},\ \delta_{\max} \triangleq \Delta_{\max} 2^{-D_l} \tag{5.8}$$

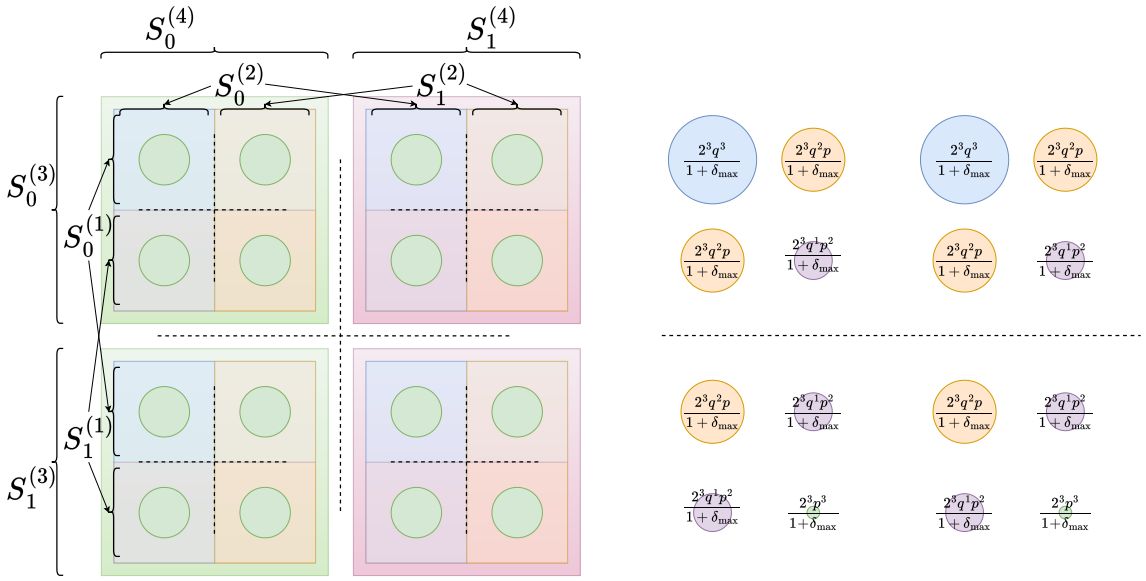

Figure 5.2: Left: Construction of the 16 "bins" used by the "look-ahead" algorithm to determine the partitions S_0 and S_1 in advance for the next four transmissions. The partitions for the l^{th} next transmission are shown by $(S_0^{(l)}, S_1^{(l)})$, $l = 1, 2, 3, 4$. The equal diameters of the green circles represents the case where posteriors $P_{\mathcal{E}_k}$ of each "bin" are equal. Right: The same partitions in the left, with the diameters modified to represent the posteriors after the receiver sees the next three transmissions that arrive at the receiver. For this example, the receiver sees three zeros. Note that the sets $(S_0^{(4)}, S_1^{(4)})$ must satisfy the "weighted median" rule, even after the posterior updates shown at the right side. This is easily satisfied in the illustration because the original bins have exactly the same probability, but in actual operation the bins will typically not be equal and the "weighted median" rule guides how different the bin probabilities can be.

The method to compute $\rho_{om}^{\min}(y^t)$ follows. Let $x_1^{D_l}(k)$ be the label of bin $\mathcal{E}_k$, and let $Z_k \triangleq \sum_{l=1}^{j} Y_{s_l+1}^{s_l+j} \oplus x_1^{D_l}(k)$. At each time $t = s_l + j$ the posterior $\rho_i(y^t)$ for $i \in \mathcal{E}_k$ will be:

$$\begin{aligned}\rho_i(y^{s_l+j}) &= \frac{\Pr(Y_{s_l+1}^{s_l+j} = y_{s_l+1}^{s_l+j} \,|\, Y^{s_l} = y^{s_l}, \theta = i)\rho_i(y^{s_l})}{\sum_{k=0}^{2^{D_l-1}} \Pr(Y_{s_l+1}^{s_l+j} = y_{s_l+1}^{s_l+j} \,|\, Y^{s_l} = y^{s_l}, \theta \in \mathcal{E}_k) P_{\mathcal{E}_k}} \\ &\geq \frac{2^j q^{j-z_k} p^{z_k} \rho_i(y^{s_l})}{1+\Delta_{\min}} \geq \frac{2^j q^{j-z_k} p^{z_k} \rho_i(y^{s_l})}{1+\Delta'_{\min}}, \end{aligned} \tag{5.9}$$

where (5.9) follows since $\{Y^{s_l} = y^{s_l}\}$ determines the partitions $\mathcal{E}_k$, $k = 0, 1, \ldots, 2^{D_l} - 1$ and $\{\theta \in \mathcal{E}_k\}$ sets $X_{s_l+1}^{s_l+j} = x_1^j(k)$. The proof of inequality (5.9) is in 5.8. A bound $\rho_{om}^{\min}(y^t)$ could

just be the smallest item on any collection $\mathcal{C} \subset \Omega$ such that $\sum_{i\in\mathcal{C}} \rho_i(y^t) \geq 1/2$. However, the largest possible lower bound $\rho^{\min}_{o_m}(y^t)$ is desired, for which the collection $\mathcal{C}$ is found using the items with largest posterior from only the "bins" U_k with larger $q^{j-z_k}p^{z_k}$. While the "bins" with larger $q^{j-z_k}p^{z_k}$ values are unknown, the number of bins sharing each value of z_k at time $t = s_l + j$ is known, and is given by $2^{D_l-j}\binom{j}{z_k}$ "bins". Thus a value of h, and a maximum z_k are chosen first, and then a value γ is found, such that:

$$\gamma 2^{-D_l} \sum_{z_k}^{h} 2^{D_l-j} \binom{j}{z_k} 2^j q^{z_k} p^{j-z_k} \geq \frac{1}{2}(1 + \Delta'_{\max}) \,. \tag{5.10}$$

Now suppose that the posteriors in each bin $\mathcal{E}_k$ are ordered such that $\rho_{o_i^k}(y^{s_l})$ is the i-th largest posterior in $\mathcal{E}_k$, and let $\rho_{o_\gamma^k}(y^{s_l})$ be the value of the posterior $\rho_{o_n^k}(y^{s_l})$ such that:

$$\sum_{i=1}^{n-1} \rho_{o_i^k}(y^{s_l}) < \gamma 2^{-D_l} \leq \sum_{i=1}^{n} \rho_{o_i^k}(y^{s_l}) \,. \tag{5.11}$$

Then, a candidate bound $\rho^{\min}_{o_m}(y^t)$ on $\rho_{o_m}(y^t)$ at time $t = s_l + j$ is given by the smallest value of $\rho_{o_\gamma^k}(y^{s_l})$, with the worst coefficient $q^{j-h}p^h$, given by:

$$\rho^{\min}_{o_m}(y^{s_l+j}) \triangleq 2^{j-h}p^h \min_{k=0,1,\ldots,2^{D_l}-1} \{\rho_{o_\gamma^k}(y^{s_l})\} \tag{5.12}$$

Two steps can help keep the right side of (5.12) as large as possible. The first is by making the smallest $\rho_{o_\gamma^k}(y^{s_l})$ the largest possible, and the second is choosing the best possible value of h. The method to make the smallest $\rho_{o_\gamma^k}(y^{s_l})$ as large as possible for a fix γ follows. Let $\rho_{o_\gamma}(y^{s_l})$ be defined in the same manner as $\rho_{o_m}(y^{s_l})$ from (5.4), but using γ instead of $\frac{1}{2}$ as follows:

$$\rho_{o_\gamma}(y^{s_l}) \triangleq \rho_{o_k}(y^{s_l}) \text{ s.t. } \sum_{i=1}^{k-1} \rho_{o_i}(y^{s_l}) < \gamma \leq \sum_{i=1}^{m} \rho_{o_i}(y^{s_l}) \,. \tag{5.13}$$

If the items i with $\rho_{o_i}(y^{s_l}) \geq \rho_{o_\gamma}(y^{s_l})$ are distributed evenly across all the "bins," then all bins

will contain at exactly $\gamma 2^{-D_l}$ and at least one bin will contain an item i with $\rho_i(y^{s_l}) = \rho_{o_\gamma}(y^{s_l})$. Since the "bins" contain an integer number of items, this might not be possible. However, a bin may be allowed to cross $\gamma 2^{-D_l}$ only when the next largest item we allocate does not fit in any other "bin," until each bin crosses $\gamma 2^{-D_l}$. Only then the smaller items are allocated in any order. At this point it still remains to guarantee that each $\delta_k \leq \delta_{\max}$, otherwise the partitioning fails.

The method to choose appropriate values of h and γ is described next. Instead of choosing h and computing γ, the value of γ is chosen first, and is used to find the smallest h that satisfies (5.10). The smallest the value of γ, the larger the value of h. Note that γ must be greater than $\frac{1}{2}$, which can by cancelling the powers of 2 in Eq. (5.10). Also note that $z_k \leq j$ since the received sequence can differ from a partial label by no more than the number j of entries in the partial label. And since

$$\sum_{z_k}^{j} \binom{j}{z_k} q^{z_k} p^{j-z_k} = 1\,, \tag{5.14}$$

then $\gamma \geq \frac{1}{2}(1 + \Delta_{\max})$ for any $\Delta_{\max}$. Using this, the "weighted median" posterior $\rho_{o_m}(y^{s_l})$ is found first and is used to compute a tentative $\Delta'_{\max}$ via (5.5). Since each bin $\mathcal{E}_k$ must have a posterior $P_{\mathcal{E}_k}$ such that $P_{\mathcal{E}_k} \geq 2^{-D_l}\gamma$, then each δ_k mus also satisfy $\delta_k \leq 2^{-D_l}(1-\gamma)$, and thus $\Delta'_{\max} \leq (1-\gamma)$. Once the value of γ and a tentative $\Delta'_{\max}$ are obtained, they are used to find the smallest h that satisfies (5.10), and then h is used to compute the lower bound $\rho^{\min}_{o_\gamma}(y^{s_l+j})$ on $\rho_{o_\gamma}(y^{s_l+j})$ via equation (5.12). This process is then repeated We using the next posterior smaller than $\rho_{o_\gamma}(y^{s_l})$. If the resulting lower bound $\rho^{\min}_{o_\gamma}(y^{s_l+j})$, is larger than the previous value, then then new value is preserved and the process is repeated. The search for the largest $\rho^{\min}_{o_\gamma}(y^{s_l+j})$ terminates when the most recently computed value is smaller than the previous value. Note that many posteriors share the same value, and since they are kept together in a group, as was done in systematic posterior matching algorithm of chapter 2, then each different posterior value $\rho_{o_i}(y^{s_l}) \geq \rho_{o_m}(y^{s_l})$, shared by an entire group, is only

tested once. This makes the search for an appropriate $\rho_{o_\gamma}^{\min}(y^{s_l+j})$ very fast. In practice, the complexity of this search is negligible compare to actually designing the partitions after finding $\rho_{o_\gamma}^{\min}(y^{s_l+j})$. Algorithm 9 describes the method to construct 2^{D_l} bins given a sorted list of groups, a tentative D and a bound $\delta_{\max}$ on the absolute difference between the bin posteriors $P_{\mathcal{E}_k}$ and the target 2^{-D} for each bean $\mathcal{E}_k$.

An example that highlights the advantage of the "weighted median" absolute difference used by the "look-ahead" algorithm is provided in Fig. 5.2. To the left, the Fig. shows the construction of 16 bins that define the partitions S_0 and S_1 for the next four transmissions. For the first transmission $S_0^{(1)}$ is comprised by the "bins" in the first and third rows and $S_1^{(1)}$ by the bins in the second and fourth rows. For the second transmission $S_0^{(2)}$ is comprised by the first and third columns and $S_1^{(2)}$ by the second and fourth columns. The last transmission, $S_0^{(4)}$ is comprised by the first and second column and $S_1^{(4)}$ by the third and fourth columns. However, just before the fourth transmission the posterior of each bins will be different, see right side of Fig. 5.2. Note that $S_0^{(4)}$ and $S_1^{(4)}$ each will contain one of the smallest updated bins, see green bins with weight $\frac{2^3p^3}{1+\delta_{\max}}$. The partitioning rules of chapter 2 must use the smallest item in $S_0^{(4)}$, which may have been attenuated by a weight approximately $\frac{2^3p^3}{1+\delta_{\max}}$. The "weighted median" absolute difference rule allows to tune the posteriors of $S_0^{(4)}$ and $S_1^{(4)}$ using the "weighted median" posterior that may lower bounded using the posteriors in bins updated with larger weights of about $\frac{2^3q^2p}{1+\delta_{\max}}$.

5.8 Proof of Thm. 6: The Weighted Median Partitioning

To prove Thm. 6 it suffices to show that the "weighted median" absolute difference rule (5.5) guarantees that constraint (3.26) is satisfied. The proofs of chapter 3 show that $|\Delta| \leq 1/3$ suffices to guarantee constraint (3.22), which may be trivially extended to the any value allowed by rule (5.5). The other constraints (3.24) and (3.25) are satisfied by the singleton rule (3.33), which is also proved in chapter 3. The proof consists of finding a lower bound on

$E[U_\theta(t+1)-U_\theta(t)|Y^t=y^t]$ that is a function of only Δ and $\rho_{o_m}(y^t)$, and then showing that the lower bound satisfies (5.15) whenever (5.5) is satisfied. Intermediate expressions, where each lower bounds the previous one, are used in the proof and the last expression will be a function of only Δ and $\rho_{o_m}(y^t)$.

Proof Thm. 6 The "Weighted Median" partitioning constraint. Need to show that:

$$\Delta^2 \leq \frac{2}{5}\rho_{o_m}(y^t) \implies E[U_\theta(t+1)-U_\theta(t)|Y^t=y^t] \geq C \tag{5.15}$$

To start, $E[U_\theta(t+1)-U_\theta(t)|Y^t=y^t]$ is expanding using the definition as follows:

$$\begin{aligned} E[U_\theta(t+1)-U_\theta(t)|Y^t=y^t] &= \sum_{i\in\Omega}\rho_i(y^t)E[U_i(t+1)-U_i(t)|Y^t=y^t,\theta=i] && (5.16)\\ &= \sum_{i\in S_0}\rho_i(y^t)E[U_i(t+1)-U_i(t)|Y^t=y^t,\theta=i] \\ &\quad + \sum_{i\in S_1}\rho_i(y^t)E[U_i(t+1)-U_i(t)|Y^t=y^t,\theta=i] && (5.17)\end{aligned}$$

The next few steps consist of expressing the random variable $U_i(t+1)$ as $U_i(t)+C+f(Y^{t+1})$, where $U_i(t)$ cancels with $-U_i(t)$. First the expectation is written in terms of $\rho_i(t)$ and $\rho_i(t+1)$, using the definition of $U_i(t)$. From Eq. (2.6) $\rho_i(t+1)$ is given by:

$$\begin{aligned}\rho_i(y^{t+1}) &= \frac{\Pr(Y_{t+1}=y_t \mid \theta=i, Y^t=y^t)\rho_i(y^t)}{\sum_{j\in\Omega}\Pr(Y_{t+1}=y_{t+1}\mid\theta=j)\rho_j(y^t)} && (5.18)\\ &= \frac{\Pr(Y_{t+1}=y_t \mid \theta=i, Y^t=y^t)\rho_i(y^t)}{\sum\limits_{j\in S_{y_{t+1}}}\Pr(Y_{t+1}=y_{t+1}\mid\theta=j)\rho_j(y^t)+\sum\limits_{j\in\Omega\setminus S_{y_{t+1}}}\Pr(Y_{t+1}=y_{t+1}\mid\theta=j)\rho_j(y^t)}\,. && (5.19)\end{aligned}$$

Since $X_{t+1}=\mathbb{1}_{\theta\in S_1}$, the indicator of the set containing θ, then:

$$\begin{aligned} j\in S_{y_{t+1}} &\implies \Pr(Y_{t+1}=y_{t+1}\mid Y^t=Y^t,\theta=j)=q \\ j\in\Omega\setminus S_{y_{t+1}} &\implies \Pr(Y_{t+1}=y_{t+1}\mid Y^t=Y^t,\theta=j)=p\end{aligned}$$

Then:

$$\rho_i(y^{t+1}) = \frac{\Pr(Y_{t+1} = y_t \mid \theta = i, Y^t = y^t)\rho_i(y^t)}{q\sum_{j\in S_{y_{t+1}}} \rho_j(y^t) + p\sum_{j\in\Omega\setminus S_{y_{t+1}}} \rho_j(y^t)} \tag{5.20}$$

When $i \in S_0$ and $Y_{t+1} = 0$ then:

$$\rho_i(y^{t+1}) = \frac{q\rho_i(y^t)}{P_0 q + P_1 p} = \frac{q\rho_i(y^t)}{\frac{1}{2} + \frac{\Delta(q-p)}{2}} = \frac{2q\rho_i(y^t)}{1+\Delta(q-p)}\,. \tag{5.21}$$

When $i \in S_0$ and $Y_{t+1} = 1$ q is replaced with p in the top of (5.21) and the sign of Δ in the bottom becomes negative. When $i \in S_1$ and $Y_{t+1} = 1$ only the sign of Δ in the bottom changes and for $i \in S_1$ and $Y_{t+1} = 0$ the only change is that q is replaced with p in the top. For $i \in S_0$, the expectation $\mathsf{E}[U_i(t+1) \mid Y^t = y^t, \theta = i \in S_0]$ is expanded using the definition (2.19) of $U_i(t)$ to obtain:

$$\mathsf{E}[U_i(t+1) \mid Y^t = y^t, \theta = i \in S_0] = q\log_2 \frac{\frac{2q\rho_i(y^t)}{1+\Delta(q-p)}}{1-\frac{2q\rho_i(y^t)}{1+\Delta(q-p)}} + p\log_2 \frac{\frac{2p\rho_i(y^t)}{1-\Delta(q-p)}}{1-\frac{2p\rho_i(y^t)}{1-\Delta(q-p)}}\,.$$

If $i \in S_1$ then, only the sign of Δ changes. Let ι_i be 1 if $i \in S_0$ and -1 if $i \in S_1$, that is: $\iota_i = \mathbb{1}_{i\in S_0} - \mathbb{1}_{i\in S_1}$. Note that the numerators $q\log_2(2q\rho_i(y^t))$ and $p\log_2(2p\rho_i(y^t)$ may be decomposed into $1 + q\log_2(q) + p\log_2(p) + \log_2(\rho_i(y^t)$, and note that the first three terms are exactly the channel capacity C. Next subtract $U_i(t) = \log_2(\rho_i(y^t)) - \log_2(1-\rho_i(y^t))$, which

is a constant given Y^t, and note that the two terms $\log_2(\rho_i(y^t)$ vanish, to obtain:

$$\mathsf{E}[U_i(t+1)-U_i(t)\,|\,Y^t=y^t,\theta=i] = q\log_2\frac{\frac{2q\rho_i(y^t)}{1+\Delta(q-p)}}{1-\frac{2q\rho_i(y^t)}{1+\Delta(q-p)}}+p\log_2\frac{\frac{2p\rho_i(y^t)}{1-\Delta(q-p)}}{1-\frac{2p\rho_i(y^t)}{1-\Delta(q-p)}} \tag{5.22}$$

$$=C+\log_2(\rho_i(y^t))-U_i(t)-\log_2(1-\rho_i(y^t))+\log_2(1-\rho_i(y^t)) \tag{5.23}$$

$$-q\log_2\big(1-\rho_i(y^t)+(q-p)(\iota_i\Delta-\rho_i(y^t)\big)-p\log_2\big(1-\rho_i(y^t)-(q-p)(\iota_i\Delta-\rho_i(y^t)\big) \tag{5.24}$$

$$= C-q\log_2\left(1+(q-p)\frac{\iota_i\Delta-\rho_i(y^t)}{1-\rho_i(y^t)}\right)-p\log_2\left(1-(q-p)\frac{\iota_i\Delta-\rho_i(y^t)}{1-\rho_i(y^t)}\right) \tag{5.25}$$

$$\geq C-\log_2\left(1+(q-p)^2\frac{\iota_i\Delta-\rho_i(y^t)}{1-\rho_i(y^t)}\right). \tag{5.26}$$

In (5.25) the argument of the logs are divided by $1-\rho_i(y^t)$ to account for the term $\log_2(1-\rho_i(y^t))$ and combine all the logs and in (5.26) Jensen's inequality is used over over p and q to lower bound (5.25).

From (5.26) the following inequality is obtained:

$$\sum_{i=1}^{M}\rho_i(y^t)\mathsf{E}[U_i(t+1)-U_i(t)\mid Y^t,\theta=i]\geq C-\sum_{i=1}^{M}\rho_i(y^t)\log_2\left(1+(q-p)^2\tfrac{\iota_i\Delta-\rho_i(y^t)}{1-\rho_i(y^t)}\right). \tag{5.27}$$

The rest of the proof consist showing that the sum in the right of (5.27) is non-negative when Δ satisfies rule (5.5). Let δ be twice the amount that the c.d.f., evaluated at the median $\rho_{o_m}(y^t)$, exceeds half, and let $R\in[0,1]$ be the fraction of posteriors greater than $\rho_{o_m}(y^t)$ that are assigned to S_0:

$$\frac{1}{2}\qquad\leq\sum_{i=1}^{m}\rho_{o_i}(y^t)=\frac{1+\delta}{2}\leq\frac{1+\rho_m(y^t)}{2} \tag{5.28}$$

$$R\frac{1+\delta}{2}=\sum_{i\in S_0}\rho_i(y^t)\mathbb{1}_{[\rho_i(y^t)\geq\rho_{o_m}(y^t)]}. \tag{5.29}$$

Then, the sum of posteriors at or above $\rho_{o_m}(y^t)$ is at least $\frac{1+\delta}{2}$. The next step consist of lower bounding the sum in the right of (5.27) by an expression independent of i, that is, a

function of $\Delta, \rho_{o_m}(y^t), \delta, R$. There are two expressions to consider, first $\frac{\Delta-\rho_i(y^t)}{1-\rho_i(y^t)}$ for the case that $\iota = 1$ and second $\frac{-\Delta-\rho_i(y^t)}{1-\rho_i(y^t)}$ for the case that $\iota = -1$. Depending on whether $i \in S_0$ or $i \in S_1$ and whether $\rho_i(y^t)$ is at least $\rho_{o_m}(y^t)$ or smaller than $\rho_{o_m}(y^t)$, one of the following four bounds will apply:

$$\rho_i(y^t) \geq \rho_{o_m}(y^t) \implies \frac{\Delta - \rho_{o_m}(y^t)}{1-\rho_{o_m}(y^t)} \geq \frac{\Delta - \rho_i(y^t)}{1-\rho_i(y^t)} \tag{5.30}$$

$$\rho_i(y^t) \geq \rho_{o_m}(y^t) \implies \frac{-\Delta - \rho_{o_m}(y^t)}{1-\rho_{o_m}(y^t)} \geq \frac{-\Delta - \rho_i(y^t)}{1-\rho_i(y^t)} \tag{5.31}$$

$$\rho_i(y^t) < \rho_{o_m}(y^t) \implies \Delta \geq \frac{\Delta - \rho_i(y^t)}{1-\rho_i(y^t)} \tag{5.32}$$

$$\rho_i(y^t) < \rho_{o_m}(y^t) \implies -\Delta \geq \frac{-\Delta - \rho_i(y^t)}{1-\rho_i(y^t)} \tag{5.33}$$

In the next steps, the following posteriors will be replaced by their equivalent expressions, in terms of Δ, δ and R.

$$\Pr\{i \in S_0 \colon \rho_i(y^t) \geq \rho_{o_m}(y^t)\} = R\frac{1+\delta}{2} \tag{5.34}$$

$$\Pr\{i \in S_0 \colon \rho_i(y^t) < \rho_{o_m}(y^t)\} = \frac{1+\Delta}{2} - R\frac{1+\delta}{2} \tag{5.35}$$

$$\Pr\{i \in S_1 \colon \rho_i(y^t) \geq \rho_{o_m}(y^t)\} = (1-R)\frac{1+\delta}{2} \tag{5.36}$$

$$\Pr\{i \in S_1 \colon \rho_i(y^t) < \rho_{o_m}(y^t)\} = \frac{1-\Delta}{2} - (1-R)\frac{1+\delta}{2} \tag{5.37}$$

The bound (5.30) will be used for the set $\{i \in S_0 \colon \rho_i(y^t) \geq \rho_{o_m}(y^t)\}$ in (5.34), the bound (5.31) will be used for the set $\{i \in S_0 \colon \rho_i(y^t) < \rho_{o_m}(y^t)\}$ in (5.35), the bound (5.32) will be used for the set $\{i \in S_1 \colon \rho_i(y^t) \geq \rho_{o_m}(y^t)\}$ in (5.36), and the bound (5.33) will be used for

the set $\{i \in S_1 : \rho_i(y^t) < \rho_{o_m}(y^t)\}$ in (5.37). The sum in (5.27) is lower bounded by:

$$
\begin{aligned}
-\sum_{i=1}^{M} \rho_i(y^t) \log_2 \left(1+(q-p)^2 \tfrac{\iota_i \Delta - \rho_i(y^t)}{1-\rho_i(y^t)}\right) \geq & - R\tfrac{1+\delta}{2} \log_2 \left(1+(q-p)^2 \tfrac{\Delta-\rho_{o_m}(y^t)}{1-\rho_{o_m}(y^t)}\right) \\
& - \left(\tfrac{1+\Delta}{2} - R\tfrac{1+\delta}{2}\right) \log_2\left(1+(q-p)^2\Delta\right) \\
& - (1-R)\tfrac{1+\delta}{2} \log_2\left(1+(q-p)^2 \tfrac{-\Delta-\rho_{o_m}(y^t)}{1-\rho_{o_m}(y^t)}\right) \\
& - \left(\tfrac{1-\Delta}{2} - (1-R)\tfrac{1+\delta}{2}\right) \log_2 \left(1-(q-p)^2\Delta\right) \qquad (5.38) \\
\geq & - \log_2 \left(1 - \tfrac{1}{2}(q-p)^2 f(\Delta, R, \delta, \rho_{o_m}(y^t))\right) , \qquad (5.39)
\end{aligned}
$$

where inequality (5.39) follows by applying Jensen's inequality over the "weights" in (5.34), (5.35), (5.36), (5.37). The terms with Δ and with $\frac{\pm\Delta-\rho_{o_m}(y^t)}{1-\rho_{o_m}(y^t)}$ are collected next to obtain:

$$
\begin{aligned}
& \Delta(-(1+\Delta)+R(1+\delta)+(1-\Delta)-(1-R)(1+\delta)) && (5.40) \\
& = \Delta((1-1-\Delta-\Delta)-(1-R-R)(1+\delta)) && (5.41) \\
& = \Delta(-2\Delta+(1-2R)(1+\delta)) && (5.42) \\
& \frac{1+\delta}{1-\rho_{o_m}}(R(\rho_{o_m}(y^t)-\Delta)+(1-R)(\Delta+\rho_{o_m}(y^t))) && (5.43) \\
& = \frac{\Delta(1+\delta)(1-2R)}{1-\rho_{o_m}(y^t)} + \frac{\rho_{o_m}(y^t)(1+\delta)(1-R+R)}{1-\rho_{o_m}(y^t)} && (5.44) \\
& = \frac{(1+\delta)(\Delta(1-2R)+\rho_{o_m}(y^t))}{1-\rho_{o_m}(y^t)} && (5.45) \\
f(\Delta,R,\delta,\rho_{o_m}(y^t)) & = \frac{\Delta(1-2R)(1+\delta)+\rho_{o_m}(y^t)(1+\delta)}{1-\rho_{o_m}(y^t)} \\
& \quad - \Delta(2\Delta-(1-2R)(1+\delta)) . && (5.46)
\end{aligned}
$$

The proof now reduces to show that $f(\Delta, R, \delta, \rho_{o_m}(y^t))$ is non negative when Δ satisfies (5.5). This only requires that the following inequality holds:

$$
\rho_{o_m}(y^t)(1+\delta) \geq 2\Delta^2(1-\rho_{o_m}(y^t)) - \Delta\rho_{o_m}(y^t)(1-2R)(1+\delta) \qquad (5.47)
$$

To remove the dependency on R in (5.47), the worst case scenario value of $\Delta(1-2R)$ is considered. For $\Delta > 0$ this happens at $R = 1$ and for $\Delta \leq 0$ at $R = 0$, so that the expression with Δ is always negative. Let $\alpha = |\Delta|$ and set $\Delta(1-2R) = \alpha$, then need:

$$\rho_{o_m}(y^t)(1+\delta)(1-\alpha) \geq 2\alpha^2(1-\rho_{o_m}(y^t)) \tag{5.48}$$

Since $0 \leq \delta \leq 1$, and $0 < (1-\alpha) < 1$, let $\delta = 0$ for a bound:

$$\rho_{o_m}(y^t)(1-\alpha) \geq 2\alpha^2(1-\rho_{o_m}(y^t)) \tag{5.49}$$

Let α^2 be bounded by a linear function of $\rho_{o_m}(y^t)$ of the form $\alpha^2 \leq \frac{a}{2b}\rho_{o_m}(y^t)$ for some $0 \leq \frac{a}{2b} < 1$. Then it is also required that:

$$\rho_{o_m}(y^t)(1-\alpha) \geq \frac{a}{b}\rho_{o_m}(y^t)(1-\rho_{o_m}(y^t)) \tag{5.50}$$

$$\alpha \leq \frac{b-a}{b} + \frac{a}{b}\rho_{o_m}(y^t) \tag{5.51}$$

To complete the proof it suffices to show that inequalities 5.50,(5.51) hold for any $\alpha^2 \leq \frac{a}{2b}\rho_{o_m}(y^t)$ with $a = 4$ and $b = 5$, where $\frac{a}{2b} = \frac{2}{5}$. For $a = 4$ and $b = 5$ then:

$$\alpha \leq \frac{1}{5} + \frac{4}{5}\rho_{o_m}(y^t) \tag{5.52}$$

Let $\rho_{o_m}(y^t) \leq \frac{1}{10}$ then $\alpha^2 \leq \frac{2}{5}\rho_{o_m}(y^t) \implies \alpha \leq \frac{1}{5}$. Now let $\frac{1}{10} \leq \rho_{o_m}(y^t) \leq \frac{49}{250}$ so that $\alpha^2 \leq \frac{49}{625} \implies \alpha \leq \frac{7}{25}$. On the right side pick the smallest $\rho_{o_m}(y^t) = \frac{1}{10}$ to obtain:

$$\alpha \leq \frac{1+4\frac{1}{10}}{5} = \frac{7}{25} \tag{5.53}$$

To show that (5.50),(5.51) hold for any $\rho_{o_m}(y^t) \in [0,1]$ the previous steps may be repeated with $\rho_{o_m}(y^t) \in [r_j, r_{j+1}]$ for $r_j = \frac{49}{250}, \frac{5}{18}, \frac{2}{5}, \frac{5}{8}, \frac{3}{4}, 1$. This completes the proof that $\alpha^2 \leq \frac{2}{5}\rho_{o_m}(y^t)$ suffices to satisfy inequality (3.26). □

Proof of inequality (5.9). To proof inequality (5.9) the expression for $\rho_i(y^{s_l+j})$ is first transformed into a function of $\rho_i(y^{s_l})$, the Hamming distance $z_{s_l,j}(i)$ between the label of the bin containing i and the feedback symbol $y_{s_l+1}^{s_l+j}$ and the bin posteriors at time $t = s_l$ and the bin probabilities $\rho_{\mathcal{E}_k}(y^{s_l})$. Then the resulting expression is lower bounded by an upper bound on the bin probabilities in the denominator as follows:

$$\begin{aligned}
\rho_i(y^{s_l+j}) = \Pr(\theta=i \,|\, Y_1^{s_l+j}=y_1^{s_l+j}) &= \frac{\Pr(\theta=i, Y_1^{s_l+j}=y_1^{s_l+j})}{\Pr(Y_1^{s_l+j}=y_1^{s_l+j})} \\
&= \frac{\Pr(Y_{s_l+1}^{s_l+j}=y_{s_l+1}^{s_l+j} \,|\, Y^{s_l}=y^{s_l},\theta=i)\rho_i(y^{s_l})}{\sum_{r\in\Omega}\Pr(Y_{s_l+1}^{s_l+j}=y_{s_l+1}^{s_l+j} \,|\, Y^{s_l}=y^{s_l},\theta=r)\rho_r(y^{s_l})} \\
&= \frac{q^{j-z_{s_l,j}(i)}p^{z_{s_l,j}(i)}\rho_i(y^{s_l})}{\sum_{k=0}^{2^{D_l}-1} q^{j-z_{s_l,j}(\mathcal{E}_k)}p^{z_{s_l,j}(\mathcal{E}_k)}\rho_{\mathcal{E}_k}(y^{s_l})} && (5.54) \\
&= \frac{q^{j-z_{s_l,j}(i)}p^{z_{s_l,j}(i)}\rho_i(y^{s_l})}{\sum_{k=0}^{2^{D_l}-1} q^{j-z_{s_l,j}(\mathcal{E}_k)}p^{z_{s_l,j}(\mathcal{E}_k)}(2^{-D_l}+\delta_k)} && (5.55) \\
&\geq \frac{q^{j-z_{s_l,j}(i)}p^{z_{s_l,j}(i)}\rho_i(y^{s_l})}{\sum_{k=0}^{2^{D_l}-1} q^{j-z_{s_l,j}(\mathcal{E}_k)}p^{z_{s_l,j}(\mathcal{E}_k)}2^{-D_l}(1+\Delta_{\max})} && (5.56) \\
&= \frac{q^{j-z_{s_l,j}(i)}p^{z_{s_l,j}(i)}\rho_i(y^{s_l})}{\sum_{k=0}^{D_l}\binom{j}{k}q^{j-k}p^k 2^{-j}(1+\Delta_{\max})} && (5.57) \\
&= \frac{2^j q^{j-z_{s_l,j}(i)}p^{z_{s_l,j}(i)}\rho_i(y^{s_l})}{1+\Delta_{\max}}\,. && (5.58)
\end{aligned}$$

In (5.55) the definition of $\delta_k = \rho_{\mathcal{E}_k}(y^{s_l}) - 2^{-D_l}$ is used, and inequality (5.56) follows from (5.8): via $2^{-D_l}\Delta_{\max} = \max_j\{\delta_j\} \leq \delta_k$. The proof is complete. □

Algorithm 8: $D, \Delta_{\max} = \textbf{computeDelta}(D, \mathbf{G})$

Input: List of Groups $\mathbf{G} = \{\mathcal{G}_0, \ldots, \mathcal{G}_{K+n_s}\}$ ▷ n_s: Number of new groups
Input: block size D ▷ number of bins: 2^D
Output: Sets $S_0, S_1, S_2, \ldots, S_{2^D-1}$ that partition $\mathbf{G}$
$target \leftarrow 2^{-D}$ ▷ Desired probability volume per bin
$S_0, S_1, \ldots, S_{2^D-1} \leftarrow \emptyset$ ▷ Initialize bins to empty
$m \leftarrow 0$, $cdf \leftarrow 0$ ▷ m: Index of first group in $\mathbf{G}$
while $cdf + N_m\rho_m < \frac{1}{2}$ **do**
  $cdf \leftarrow cdf + N_m\rho_m(t)$ ▷ $N_m, \rho_m \in \mathcal{G}_m$
  $m \leftarrow m + 1$ ▷ Increase group index m
end
$\Delta_{temp} = -1$, $\Delta = -2$, $thd \leftarrow 1$
while $\Delta_{\max} < \Delta_{temp}$ **do**
  $\Delta_{\max} \leftarrow \Delta_{temp}$, $\Delta_{temp} \leftarrow \frac{2\times cdf-1}{2\times cdf+1}$
  $s \leftarrow 1$ ▷ Need: $\frac{1-\Delta'}{1+\Delta'} s \times cdf \geq \frac{1}{2}$
  for $a = 0, \ldots, D-1$ **do**
    $W' \leftarrow (2q)^a (2p)^{D-a-1}$
    $\Delta' \leftarrow \min\{\frac{2s\times cdf-1}{2s\times cdf+1}, \sqrt{\frac{2}{5}\rho_m(t)W'}\}$
    $W_m^a \leftarrow \frac{(2q)^{D-a}(2p)^{a-1}}{1+\Delta'}$
    $\Delta' \leftarrow \min\{\Delta', \sqrt{\frac{2}{5}\rho_m(t)W_m^a}\}$
    if $\Delta' < \Delta_{temp}$ **then**
      $thd \leftarrow 2^{-D} cdf$
      break
    end
    $\Delta_{temp} \leftarrow \Delta'$
    $s \leftarrow s - \binom{D-1}{a} q^a p^{D-1-a}$
  end
  $cdf \leftarrow cdf + N_m\rho_m(t)$
  $m \leftarrow m + 1$
end
if $\Delta_{\max} < 0$ **then**
  $D \leftarrow D - 1$
  Repeat process
end

Algorithm 9: $(S_0, S_1, S_2, \ldots, S_{2^D-1}) =$ **PartitionGroups**$(\mathbf{G})$

Input: List of Groups $\mathbf{G} = \{\mathcal{G}_0, \ldots, \mathcal{G}_{K+n_s}\}$ ▷ n_s: Number of new groups
Input: block size D, threshold $\delta_{\min}$ ▷ The number of bins is 2^D
Output: Sets $S_0, S_1, S_2, \ldots, S_{2^D-1}$ that partition $\mathbf{G}$
$target \leftarrow 2^{-D}$ ▷ Desired probability volume per bin
$S_0, S_1, \ldots, S_{2^D-1} \leftarrow \emptyset$ ▷ Initialize bins to empty
$\delta_0, \delta_1, \ldots, \delta_{2^D-1} \leftarrow target$ ▷ Initialize bins to empty
$cdf \leftarrow 0$ ▷
$m \leftarrow 0$ ▷ Index of first group in $\mathbf{G}$
$\rho_s \leftarrow \rho_m$ ▷ Initial median
$temp \leftarrow \mathcal{G}_0$
for $s = 1, \ldots, D-1$ **do**
  $f_0, f_1, \ldots, f_{2^D-1} \leftarrow False$ ▷ Flag items passed threshold
  while $crossBins > 0$ **do**
    Distribute items evenly across bins
    for $j = 0, \ldots, 2^D - 1$ **do**
      if f_0 *and* $P_j > thd_s$ **then**
        $f_j \leftarrow True$ ▷ Avoid double counting bins
        $\rho_s \leftarrow \min\{\rho_s, W_s^+ \cdot temp \rightarrow \rho(y^t)\}$ ▷ min possible next median
        $crossBins \leftarrow crossBins - 1$
      end
    end
    $temp \leftarrow temp \rightarrow Next$ ▷ Move to next item
  end
end
$f_0, f_1, \ldots, f_{2^D-1} \leftarrow False$ ▷ Flag items passed threshold
while *True* **do**
  Distribute items evenly across empty bin space
  $temp \leftarrow temp \rightarrow Next$ ▷ Move to next item
  for $i = 0, \ldots, 2^D - 1$ **do**
    if $P(S_i) \geq 2^{-D}$ **then**
      $f_i = 1$ ▷ Set bin j to done
    end
  end
end
for $i = 0, \ldots, 2^D - 1$ **do**
  Assert $\mid P(S_j) - 2^{-D} \mid \leq 2^{-D}\delta_{\min}$
end

5.9 The Sparse "Sub-block Combining" Algorithm

This section explores a variant of the sub-block combining algorithm as a sparse feedback encoder. The "look-ahead" algorithm allows to enforce constraints that guarantee the rate performance of sequential SPM algorithm with significant feedback sparsity. However, the complexity of partitioning with the "look-ahead" algorithm increases exponentially with the delay factor D. This is because the number of partitions, or bins, built before each block transmission is given by 2^D. This exponential complexity limits further sparsity. One method that would allow to reduce the complexity, or keep increasing the sparsity as the message length grows, is to segment the message and operate on two segments at the same time. This approach is not unlike the "sub-block combining" algorithm of 4. This dissertation proposes to modify the "sub-block combining" algorithm to operate in all sub-blocks simultaneously, for as long as permitted by system constraints. This modification results in a sparse "sub-block combining" algorithm.

The sparse "sub-block combining" algorithm segments the message into an initial number $D_l = D_0$ of pieces, (usually the same size, but not necessary), and constructs a posterior matching system for each message segment. The transmitter sends one symbol per message segment, and the feedback symbols are stored at the receiver until the symbol for the last segment is received. Then, all D_l feedback symbols are sent back to the transmitter, and the process repeats. For each sub-block system j, encoder and decoder partition the candidate segments into two sets, $S_0^{(j)}$ and $S_1^{(j)}$, $j = 1, 2, \ldots, D_l$, according to some partitioning rule. This is done before transmitting the first symbol X_{s_l} of the D_l symbols $X_{s_l}, X_{s_l+1}, \ldots, X_{s_l+D_l}$. If the partitioning rule cannot be satisfied, pairs of sub-blocks are combined into a single larger sub-block, not unlike the "causal" version of the algorithm. However, it is not necessary to maintain the segments in order. The algorithm can choose the pairs to maximize the probability that the partitioning rule is satisfied with the minimum number of pair combinations. The number D_{l+1} of blocks that remain when the partitioning rule is satisfied,

depends on the state of the system, and the partitioning rule itself.

The partitioning rule that the sparse "sub-block combining" algorithm enforces could be selected to satisfy the "Weighted Median partitioning" constraint that guarantees the same rate performance that the non-sparse SPM algorithm guarantees. For this rule, a bound on the "weighted median" needs to be computed. An outline of the procedure to determine the balance of the partitions follows: First find a worst-case-scenario update on each previous block and compute the corresponding "weighted median" value within each partition $S_0^{(j)}, S_1^{(j)}$. Compute the overall "weighted median" as the product of all the "weighted medians" of each sub-block, and use it to compute an initial Δ'. Then, use the initial Δ' to estimate a lower bound on the update coefficient of each "weighted median, and use it to compute a new estimate Δ. Finally, select the minimum between Δ' and Δ to determine the maximum difference between the partitions at each sub-block.

The sparse "sub-block combining" algorithm has advantages and disadvantages. The main advantage is the lower complexity. Only two partitions, or "bins" are needed for each sub-system. To transmit D_l bits, the sparse "sub-block combining" needs to construct 2 partitions per sub-block, for a total of $2 \cdot D_l$ partitions: $(S_0^{(1)}, S_1^{(1)}), (S_0^{(2)}, S_1^{(2)}), \ldots, (S_0^{(D_l)}, S_1^{(D_l)})$. The "look-ahead" algorithm needs 2^{D_l} partitions to achieve the same sparsity, placing a severe restriction on feasible values of D_l. The lower complexity is traded for two main disadvantages: first, the value of D_l decreases monotonically. That is, once two sub-blocks are combined, they are never split again, and a single symbol is transmitted for that sub-block. The other disadvantage is that the randomness of the overall process is not distributed uniformly across all the sub-blocks. Some posteriors in a sub-block could grow past the point that allows to satisfy the partitioning rule, and will be combined with others, while the posteriors of other sub-blocks could be far from the point when they need to be combined. A single symbols will be transmitted for each sub-block that remains after combining the necessary pairs. Instead, the "look-ahead" algorithm computes D_l based on the entire process. That is, the posteriors on some sub-blocks could be large, while and still the posteriors of the larger message be

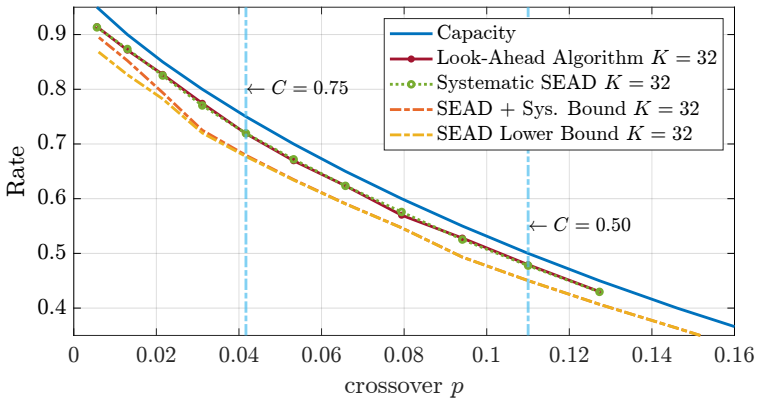

Figure 5.3: Rate performance vs. channel p of the look-ahead algorithm for $K = 32$. The solid solid dark blue curve shows the channel capacity. The "look-ahead algorithm" curve is the brown solid line $-\circ$. The green solid line $-\circ$ is for the non-sparse algorithm in Ch. 2. The orange line dash is the rate lower bound $K/\mathsf{E}[\tau]$ for systematic transmission using (3.27) and the yellow dash line is the lower bound from (3.34) for uniform input distribution.

all small enough to allow a larger value of D_l with the "look-ahead" algorithm, that the "sub-block combining" algorithm will not be able to achieve.

5.10 Simulation Results

Simulation results for the "look-ahead" algorithm and the sparse "sub-block combining" algorithm demonstrate how "sparse" the feedback times can be while still achieving the desired reliability and maintaining a rate above the bounds for the non-sparse case. Fig. 5.3 provides the rate performance of the look-ahead algorithm (The red curve with $\bullet$ markers labeled "Look-Ahead algorithm") vs. the crossover probability p for $K = 32$ message bits. Fig. 5.3 validates Thm. 4, showing that the "look-ahead" algorithm using the "weighted median" absolute difference rule achieves rates that exceeds even the SEAD lower bounds on rate from (3.34) and (3.398) on the expected blocklength $\mathsf{E}[\tau]$, which are above the bound of Thm. 4, which is derived from (3.34) that is provided in Thm. 6. The performance of the systematic posterior matching algorithm of Ch. 2, the green dotted curve with $\circ$ markers labeled "Systematic SEAD" is provided in Fig. 5.3 for reference. Note that the rate of the "look-ahead"

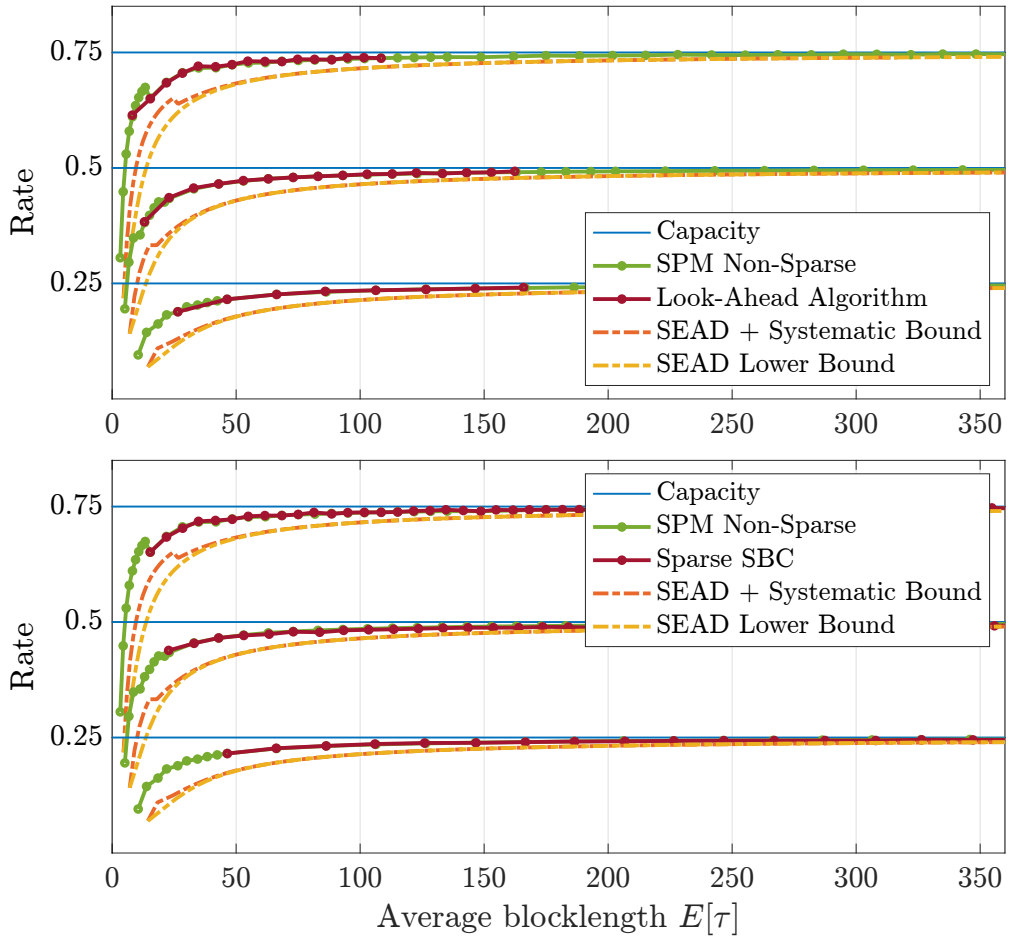

Figure 5.4: Rate performance of the "look-ahead" algorithm (top sub-plot) and the "sub-block combining" algorithm (bottom sub-plot) vs. average blocklength $\mathsf{E}[\tau]$. The curves are over channels with capacities $C = 0.25$, $C = 0.50$ and $C = 0.75$ shown with the horizontal solid blue lines. The rate performance of the "look-ahead" and the "sub-block combining" algorithms are shown with the red solid lines with dots. The green solid line with dots are for the non-sparse SPM algorithm of Ch. 2. The orange dashed curves are the rate lower bound $K/\mathsf{E}[\tau]$ for for systematic transmission from (3.27) and the yellow dash dot lines are for the lower bound from (3.34) for uniform input distribution.

algorithm is indistinguishable from that of the "Systematic SEAD" algorithm, which satisfies the highest achievability bounds in this dissertation.

The rate performance vs. average blocklength $\mathsf{E}[\tau]$ of the "look-ahead" algorithm is provided in the top plot of Fig. 5.4 (red solid curve with $\circ$ markers labeled "Look-Ahead Algorithm"), and the rate performance of the sparse "sub-block combining" algorithm is provided

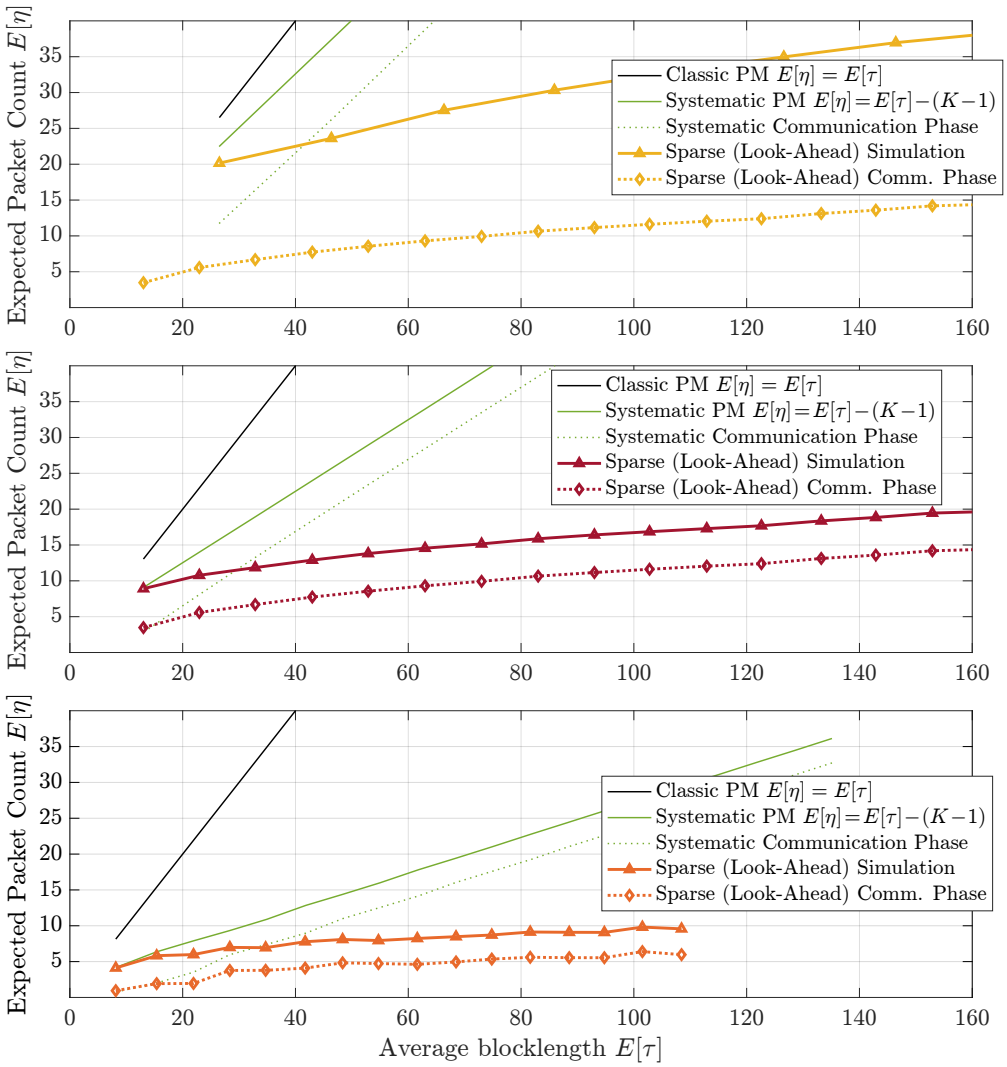

Figure 5.5: Sparsity performance of the "look-ahead" algorithm in expected feedback packet count $\mathsf{E}[\eta]$ vs. average blocklength $\mathsf{E}[\tau]$ (the solid lines with Δ markers) for a channels with capacities $C = 0.25$, $C = 0.50$ and $C = 0.75$. The dotted lines with $\Diamond$ markers only count feedback packets for the communication phase, the target of the "look-ahead" algorithm. The yellow curves in the top plot are for the "look-ahead" algorithm over a channel with capacity $C = 0.50$, and orange curves in the bottom plot are for the "look-ahead" algorithm over a channel with capacity $C = 0.75$. The solid black lines labeled "Classic PM $E[\eta] = \mathsf{E}[\tau]$" represent classical PM with feedback after every symbol. The solid green lines labeled "Systematic PM $\mathsf{E}[\eta] = \mathsf{E}[\tau] - (K-1)$ represent the expected feedback packet count $\mathsf{E}[\eta]$ for systematic PM, which avoids the first $K-1$ feedback transmissions. The dotted green lines labeled "Systematic Communication Phase" only counts feedback packets for the target communication phase of systematic PM.

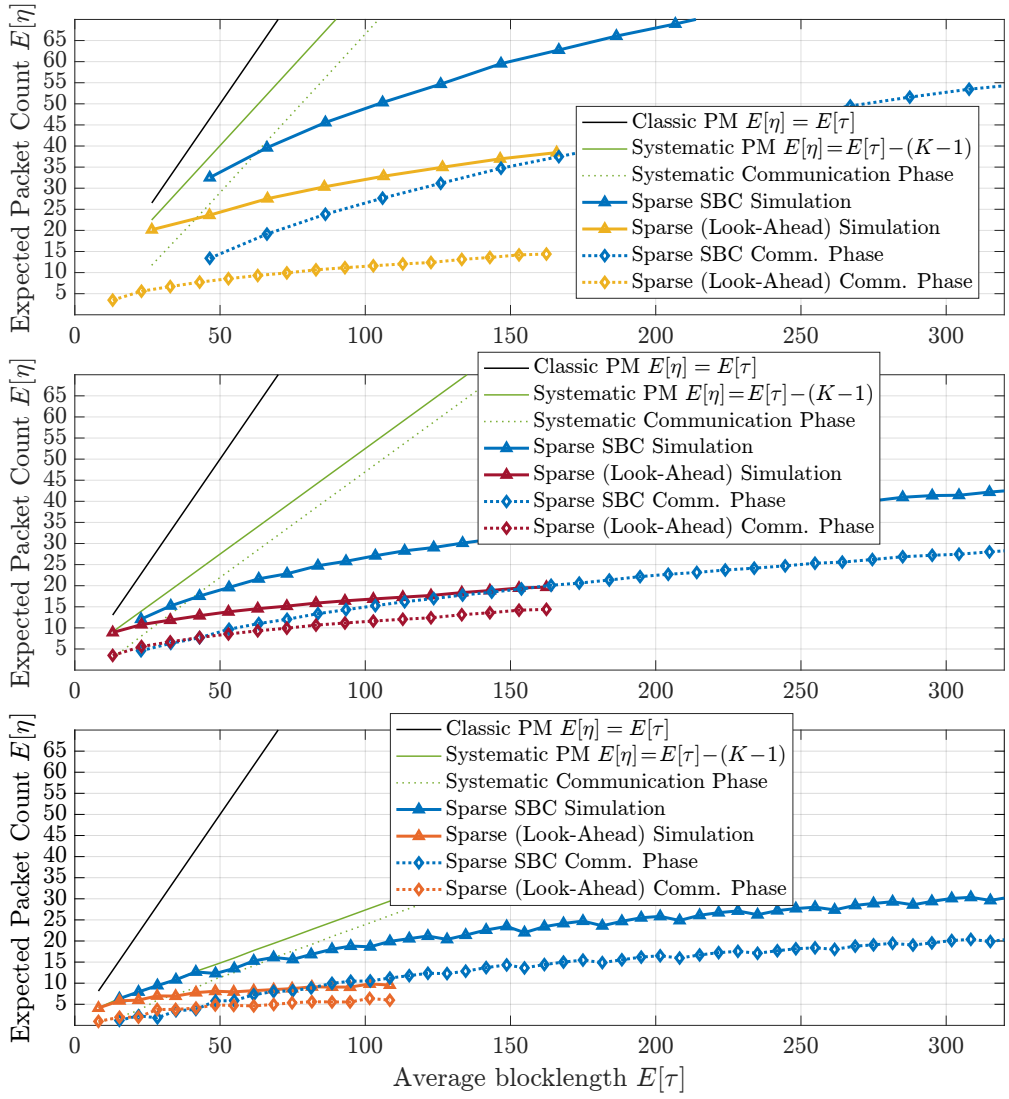

Figure 5.6: Sparsity performance in terms of expected feedback packet count $\mathsf{E}[\eta]$ vs. average blocklength $\mathsf{E}[\tau]$ of the sparse "sub-block combining" algorithm (blue the solid lines with Δ markers) is provided, in addition to the curves of Fig. 5.5. The Fig. compares the sparsity performance of the "look-ahead algorithm" and the sparse "sub-block combining" algorithm. The top plot is for a channel with capacity $C = 0.25$, the middle plot is for a channel with capacity $C = 0.50$ and the bottom plot is for a channel with capacity $C = 0.75$ The blue dotted lines with $\Diamond$ markers only count feedback packets for the communication phase, also the target of the sparse "sub-block combining" algorithm.

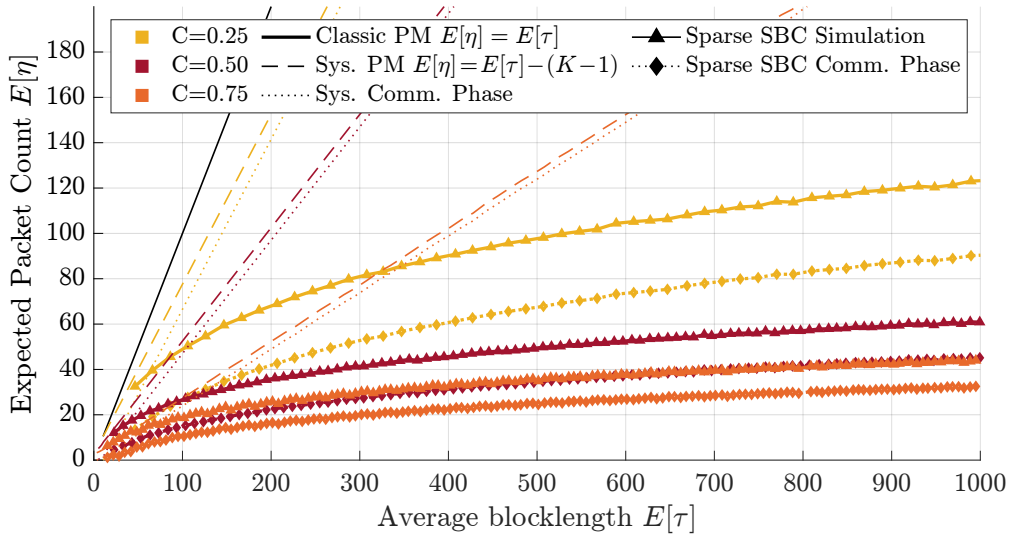

Figure 5.7: Feedback times sparsity performance of the sparse 'sub-block combining" algorithm as expected feedback packet count $\mathsf{E}[\eta]$ vs. average blocklength $\mathsf{E}[\tau]$ over channels with capacity $C = 0.25$, $C = 0.50$ and $C = 0.75$. The curves are the same provided in Fig. 5.6 with the blue lines, but the horizontal axis extends to $\mathsf{E}[\tau] = 1000$ to highlight larger message sizes that can be transmitted by the "sub-block combining" algorithm because of a much lower complexity than the "look-ahead" algorithm.

in the bottom plot of Fig. 5.4 (red solid curve with $\circ$ markers labeled "Sparse SBC"). Fig. 5.4 also includes the rate of the SPM algorithm from Ch. 2 and the bounds described in Fig. 5.3. The rate curves validate the claims of Thm. 4 regarding the rate of an encoder, like the "look-ahead" algorithm that satisfies the "weighted median" absolute difference rule.

The "sparsity" performance of the "look-ahead" algorithm in terms of expected feedback packet count $\mathsf{E}[\eta]$ vs. expected blocklength $\mathsf{E}[\tau]$ is provided in Fig. 5.5 for channels with capacity $C = 0.25$, $C = 0.50$ and 0.75. Fig. 5.5 also includes the expected packet count restricted to the non-systematic transmissions of the communication phase (see dotted lines with $\Diamond$ markers labeled "Sparse (Look-Ahead) Comm Phase") to compare the sparsity in the communication phase targeted by the algorithm. For reference, Fig. 5.5 includes the expected packet counts of classical posterior matching, with feedback after every symbol, and of the systematic posterior matching schemes in Ch. 2 that may use a single feedback packet for all the systematic transmissions.

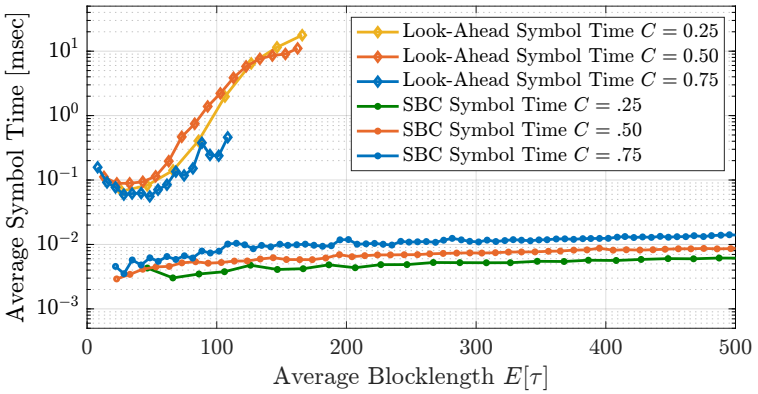

Figure 5.8: Average runtime per transmitted symbol, in milliseconds, vs. average blocklength $\mathsf{E}[\tau]$ of the "look-ahead" algorithm and the sparse "sub-block combining" algorithm over channels with capacity $C = 0.25$, $C = 0.50$ and $C = 0.75$. The top three curves with $\Diamond$ markers are for the "look-ahead" algorithm and the bottom three curves with $\circ$ markers are for the "sub-block combining" algorithm. The plots are for simulations of 10000 trials at each point.

The performance of the sparse "sub-block combining" algorithm in terms of expected feedback packet count $\mathsf{E}[\eta]$ vs. expected blocklength $\mathsf{E}[\tau]$ is compared to that of the "look-ahead" algorithm in Fig. 5.6. The simulations show an average number of feedback packets that grows slowly with the blocklength, but still require a higher number of feedback packets than the "look-ahead" algorithm. When compared to feedback after every symbol, the sparse "sub-block combining" algorithm exhibits good performance, with an average number of feedback packets that grows slowly as the blocklength increase. The performance of the "sub-block combining" algorithm at larger blocklengths is better illustrated in Fig. 5.6

The average number of feedback packets $\mathsf{E}[\eta]$ vs. the average blocklength $\mathsf{E}[\tau]$ for the sparse "sub-block combining" algorithm, over a wider blocklength range up to 1000 bits is shown in Fig. 5.7. The plots of Fig. 5.7 are for the same simulations of the "sub-block combining" algorithm in Fig. 5.6, over channels with capacity $C = 0.25$, $C = 0.50$ and $C = 0.75$. The lower complexity of the "sub-block" combining algorithm allows sparse feedback communication over much larger blocklengths than the "look-ahead" algorithm, which becomes too complex, even for simulations, at blocklengths between $100 - 160$ bits.

Note that for a channel with capacity $C = 0.50$ and average blocklength of $\mathsf{E}[\tau] = 1000$ (see red curves with Δ and $\Diamond$ markers in Fig. 5.7), the "sub-block" combining algorithm uses about 45 feedback packet in the communication phase, while a systematic algorithm like the SPM requires about 500 feedback packets during the communication phase. Thus the sub-block combining algorithm achieves an average of 10 bits between feedback transmissions when when the expected blocklength $\mathsf{E}[\tau]$ is about 1000.

The average runtime per transmitted symbol of the "look-ahead" and the "sub-block combining" algorithms (in milliseconds) vs. the average blocklength $\mathsf{E}[\tau]$ is shown in Fig. 5.8. The curves show that the runtime of the "look-ahead" algorithm increases very rapidly with increasing blocklength, which makes it impractical for blocklengths little above 100 bits. While the curves for the "look-ahead" algorithm seem to taper down as blocklength increases, this is probably artifact introduced by a cap on the largest D_l, which we set at $D_l \leq 10$ because of hardware and memory restrictions. This cap on D_l prevents the "look-ahead" algorithm from improving further as the blocklength increases. In contrast, the runtime curves for the sparse "sub-block combining" algorithm show a much slower increase with blocklength and demonstrate its lower complexity, compared to the "look-ahead" algorithm. This lower complexity allows to efficiently transmit much larger messages and blocklenghts up to 1000 bits, as shown in Fig. 5.7.

5.11 Conclusion

This chapter explores how well posterior matching with noiseless feedback of received symbols can perform with sparse feedback times instead of having feedback after every symbol. The chapter begins by showing how a "look-ahead" algorithm can limit the frequency of feedback transmissions while still obtaining the achievable rates of posterior matching with feedback after every symbol. No feedback is required until after the initial systematic transmission of the message bits. After that, a new weighted median absolute difference rule facilitates

partitioning that allows multiple symbols to be transmitted before feedback is required for a new partitioning step.

The complexity of the "look-ahead" algorithm grows rapidly with message size. As an alternative, this chapter proposes a sparse "sub-block combining" algorithm, which exhibits much lower complexity while still achieving good sparsity performance for larger message sizes. While the "sub-block combining" algorithm cannot achieve the sparsity of the "look-ahead" algorithm for short message sizes, the lower complexity of the sparse "sub-block combining" algorithm allows sparse operation on larger message sizes and lower capacity channels for which the "look-ahead" algorithm requires too much complexity.

Future research could combine features of both algorithms to manage the complexity of the 'look-ahead' algorithm. Each sub-block of the "sub-block" combining algorithm could be a "look-ahead" system that has a manageable number of bins. The number of sub-blocks could be chosen to keep the number of "bins" at each sub-block low enough to maintain low complexity, but not restricted to just two, as with the current "sub-block" combining algorithm. For example, five symbols per sub-block would require 32 bins per sub-block. Then, the total number of bins will be the number of sub-blocks, times 32. If there are N sub-blocks, the number of symbols transmitted before feedback is needed will be up to $5N$, while the required number of partitions will be at most $32 \times N$. In contrast, using the "look-ahead" algorithm alone would require 2^{5N} bins to achieve the same sparsity. The current "sub-block combining" algorithm alone could only transmit N symbols before feedback is needed.

CHAPTER 6

Conclusion

This dissertation investigates communication over the binary symmetric channel, aided by a noiseless feedback channel. For the systems described in Chapters 2 and 3, the transmitter has full knowledge of both the source message and the decoder state. The only constraint is a small upper bound on the error rate, or the probability of incorrectly decoding the transmitted message. In Chapter 4, the encoder has an additional constraint imposed by the information source. The information sequence is not available to the encoder at the start of transmission; the information source makes the information sequence causally available as the transmission progresses. Finally, Chapter 5 investigates the delay of the feedback symbols. The transmitter seeks to maintain a rate lower bounded by bounds derived in chapter 3 for the sequential case, while minimizing the number of instances where the feedback is relayed from the receiver.

Chapter 2 Provides efficient algorithms and a simulation framework that implements communication over the binary symmetric channel with noiseless feedback. This chapter proposes a very simple encoding rule which only requires finding the weighted median of the posteriors, and avoids most of the operations needed for other encoding rules like the SED. Simulations of the algorithms validate the previously proposed analytical bounds, as well as the ones introduced in this dissertation. The algorithms exhibit very low runtime complexity, that is quadratic in order but dominated by the linear term, for the first few thousand bits. This is the region where variable-length feedback codes provide a meaningful advantage over forward error correcting codes.

Chapter 3 provides a new analysis to prove the achievability bounds proposed by Yang *et al*, [YPA21]. The new analysis admits a tighter bound and requires looser constraints than those in [YPA21]. Particularly, the new looser constraints are satisfied by the SEAD encoder with thresholding of order posteriors partitioning. The chapter generalizes the surrogate process analysis of Yang *et al.*, to provide applications to a broader class of processes that meet certain constraints. The constraints are proven for the SEAD encoder, thus proving that the SEAD encoder satisfies the same achievability bound than the SED encoder. A new converse bound is also proposed in Chapter 3, that applies to any encoder that enforces the stopping rule (2.1).

Chapter 4 explores causal encoding where the decoding time is of the essence and the source information sequence is made available to the encoder causally as the transmission progresses. The lower and upper bounds on decoding times are identified, as well as the regions where causal encoders could provide lower expected decoding time by avoiding the limitations of non-causal schemes. The chapter proposes a sub-block combining algorithm, a causal encoding scheme that outperforms non-causal schemes across the entire region where they are limited by the source constraints. Finally the chapter presents an analysis that optimizes block sizes matched to each region of operation. The optimized block sizes further reduce the expected decoding time of the sub-block combining algorithm.

Chapter 5 explores sparsity in the feedback transmission times. Instead of sending the feedback symbols to the encoder as soon as they are received, the decoder may choose to wait for a few symbols to accumulate, and later transmit them in a single packet. The chapter proposes a weighted median absolute difference rule, a partitioning rule that relaxes the SEAD constraints in Chapter 3 while also satisfying the constraints of Thm. 1, thus guaranteeing the rate performance derived in the Thm. The new rule allows a portion of the transmission to encode more than one symbol in advance, without receiving additional feedback. The look-ahead algorithm is provided to enforce the constraints and maximize feedback sparsity. The look-ahead algorithm works well for lower values of message size K.

Finally the sub-block combining algorithm of Chapter 4 is modified as a sparse feedback encoder, by operating on several sub-blocks simultaneously. This version of the sub-block combining algorithm exhibits much lower complexity than the look-ahead algorithm. For low message sizes the look-ahead algorithm provides superior sparsity performance, but the complexity quickly limits the sparsity that can be achieved. The low complexity of the sparse sub-block combining algorithm makes it suitable for much larger block sizes, where it is able to achieve more sparsity than the look-ahead algorithm.

6.1 Future Research Directions

Future research directions on the topics covered in this dissertation are outlined below. In the converse bound 4 of chapter 3, a more rigorous proof is needed for the expression to bound $\tau_0 + S$ in Eq. (3.129) and Eq. (3.127), that accounts for the case where the log of the posterior of the sampled message θ never crosses the zero threshold. The current proof is more a sketch than a rigorous mathematical proof. Another interesting research direction could be to find the class of system to which the converse bound generalizes to; for instance, whether the converse bound applies to any system that attains an error rate bounded by the same ϵ.

For causal encoding, the sub-block combining algorithm may be modified to use dynamic block sizes that depend on the duration of the communication phase for each previous sub system. In the sparse feedback times problem, a surrogate process like the one used to prove Thm. 3 for the "thresholding of ordered posteriors" rule remains an open problem for the "weighted median absolute difference" rule. If such process is found, all bounds introduced in chapter 3 would also apply to an encoder that enforces the "weighted median" absolute difference rule. The simulation results of chapter 5 indicate that this surrogate process may exist. Sparse feedback times in the confirmation phase is another topic that is left for future research. The sparse feedback times problem could also be studied under different problem

formulations, like relaxing the rate constraint to obtain further sparsity of the feedback times. Finally, the problem of communication with noisy feedback remains an open area of research and may be suitable for practical applications.

CHAPTER 7

Notations and Definitions

The following definitions are used throughout this dissertation, and may also be found before they are used in each section. They are also included here for convenience.

Posterior probability:	$\rho_i(y^t)$	$\triangleq P(\theta = i \mid Y^t = y^t),\ \forall i \in \{0,1\}^K$	(7.1)
Log likelihood ratio:	$U_i(t)$	$\triangleq \log_2\left(\dfrac{\rho_i(Y^t)}{1-\rho_i(Y^t)}\right)\ \forall i \in \{0,1\}^K$	(7.2)
Binary entropy	$H(p)$	$\triangleq -p\log_2(p) - (q)\log_2(q)\,, \quad q \triangleq 1-p$	(7.3)
Channel capacity:	C	$\triangleq 1 - H(p)$	(7.4)
Phase II step size	C_2	$\triangleq \log_2\left(\dfrac{q}{p}\right)$	(7.5)
Phase II average step size	C_1	$\triangleq q\log_2\left(\dfrac{q}{p}\right) + p\log_2\left(\dfrac{p}{q}\right) = (q-p)C_2$	(7.6)
Original stopping time	τ	$\triangleq \min_{t\in\mathbb{N}}\{\exists i \in \Omega : \rho_i(y^t) \geq 1-\epsilon\}$	(7.7)
Alternative stopping time	τ	$\triangleq \min_{t\in\mathbb{N}}\left\{\exists i \in \Omega : U_i(t) \geq \left\lceil \dfrac{\log_2\left(\frac{1-\epsilon}{\epsilon}\right)}{C_2} \right\rceil C_2\right\}$	(7.8)
SED rule		$0 \leq \sum_{i\in S_0}\rho_i(y^t) - \sum_{i\in S_1}\rho_i(y^t) < \min_{i\in S_0}\rho_i(y^t)$	(7.9)
SEAD rule		$\left\lvert \sum_{i\in S_0}\rho_i(y^t) - \sum_{i\in S_1}\rho_i(y^t)\right\rvert \leq \min_{i\in S_0}\rho_i(y^t)$	(7.10)
Posterior of S_0	P_0	$\triangleq \sum_{i\in S_0}\rho_i(y^t)$	(7.11)
Posterior of S_1	P_1	$\triangleq \sum_{i\in S_1}\rho_i(y^t)$	(7.12)
Partition difference	Δ	$\triangleq \sum_{i\in S_0}\rho_i(y^t) - \sum_{i\in S_1}\rho_i(y^t)\,.$	(7.13)

REFERENCES

[AGW23] Amaael Antonini, Rita Gimelshein, and Richard D. Wesel. "Achievable Rates for Low-Complexity Posterior Matching over the Binary Symmetric Channel." *IEEE Transactions on Information Theory*, 2023.

[AJB10] Abdullah Akce, Miles Johnson, and Timothy Bretl. "Remote teleoperation of an unmanned aircraft with a brain-machine interface: Theory and preliminary results." In *2010 IEEE Int. Conf. Robotics and Autom.*, pp. 5322–5327, 2010.

[AJD13] Abdullah Akce, Miles Johnson, Or Dantsker, and Timothy Bretl. "A Brain–Machine Interface to Navigate a Mobile Robot in a Planar Workspace: Enabling Humans to Fly Simulated Aircraft With EEG." *IEEE Trans. Neural Syst. Rehabil. Eng.*, **21**(2):306–318, 2013.

[Ana12] Achilleas Anastasopoulos. "A sequential transmission scheme for unifilar finite-state channels with feedback based on posterior matching." In *2012 IEEE Int. Symp. Inf. Theory*, pp. 2914–2918, 2012.

[AYW20] A. Antonini, H. Yang, and R. D. Wesel. "Low Complexity Algorithms for Transmission of Short Blocks over the BSC with Full Feedback." In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 2173–2178, 2020.

[BA10] Jung Hyun Bae and Achilleas Anastasopoulos. "A posterior matching scheme for finite-state channels with feedback." In *2010 IEEE Int. Symp. Inf. Theory*, pp. 2338–2342, 2010.

[Bur76] M. V. Burnashev. "Data Transmission over a Discrete Channel with Feedback. Random Transmission Time." *Problemy Peredachi Inf.*, **12**(4):10–30, 1976.

[CRJ19] Sung-En Chiu, Nancy Ronquillo, and Tara Javidi. "Active Learning and CSI Acquisition for mmWave Initial Alignment." *IEEE J. Sel. Areas Commun.*, **37**(11):2474–2489, 2019.

[CYK23] James Y. Chen, Recep Can Yavas, and Victoria Kostina. "Variable-Length Codes with Bursty Feedback." In *2023 IEEE Intl. Symp. on Inf. Theory (ISIT)*, pp. 1448–1453, 2023.

[Dur19a] Richard Durrett. *Probability : theory and examples / Chapter 4.2, Exponential Martingale.* Cambridge series in statistical and probabilistic mathematics ; 49. Cambridge University Press, Cambridge ;, fifth edition, 2019.

[Dur19b] Richard Durrett. *Probability : theory and examples / Chapter 4.8, Optional Stopping Theorems.* Cambridge series in statistical and probabilistic mathematics ; 49. Cambridge University Press, Cambridge ;, fifth edition, 2019.

[GC10] Siva K. Gorantla and Todd P. Coleman. "A stochastic control approach to coding with feedback over degraded broadcast channels." In *49th IEEE Conference on Decision and Control (CDC)*, pp. 1516–1521, 2010.

[GC11] Siva K. Gorantla and Todd P. Coleman. "Information-Theoretic Viewpoints on Optimal Causal Coding-Decoding Problems." *CoRR*, **abs/1102.0250**, 2011.

[Hor63] M. Horstein. "Sequential transmission using noiseless feedback." *IEEE Transactions on Information Theory*, **9**(3):136–143, July 1963.

[KD14] Ashish Khisti and Stark Draper. "The Streaming-DMT of Fading Channels.", 2014.

[KMM13] Sanggyun Kim, Rui Ma, Diego Mesa, and Todd P. Coleman. "Efficient Bayesian inference methods via convex optimization and optimal transport." In *2013 IEEE Int. Symp. on Inf. Theory*, pp. 2259–2263, 2013.

[KPV17] Victoria Kostina, Yury Polyanskiy, and Sergio Verd. "Joint Source-Channel Coding With Feedback." *IEEE Trans. on Inf. Theory*, **63**(6):3502–3515, 2017.

[LG15] C. T. Li and A. El Gamal. "An Efficient Feedback Coding Scheme With Low Error Probability for Discrete Memoryless Channels." *IEEE Transactions on Information Theory*, **61**(6):2953–2963, June 2015.

[LKJ20] Anusha Lalitha, Anatoly Khina, and Tara Javidi. "Causal Posterior Matching and its Applications.", 2020.

[LMS07] Christopher Lott, Olgica Milenkovic, and Emina Soljanin. "Hybrid ARQ: Theory, State of the Art and Future Directions." In *2007 IEEE Inf. Theory Workshop on Information Theory for Wireless Networks*, pp. 1–5, 2007.

[LTK15] Si-Hyeon Lee, Vincent Y. F. Tan, and Ashish Khisti. "Streaming Data Transmission in the Moderate Deviations and Central Limit Regimes.", 2015.

[NJW15] M. Naghshvar, T. Javidi, and M. Wigger. "Extrinsic Jensen–Shannon Divergence: Applications to Variable-Length Coding." *IEEE Transactions on Information Theory*, **61**(4):2148–2164, April 2015.

[NWJ12] M. Naghshvar, M. Wigger, and T. Javidi. "Optimal reliability over a class of binary-input channels with feedback." In *2012 IEEE Inf. Theory Workshop*, pp. 391–395, Sep. 2012.

[PPV10] Yury Polyanskiy, H. Vincent Poor, and Sergio Verdu. "Channel Coding Rate in the Finite Blocklength Regime." *IEEE Transactions on Information Theory*, **56**(5):2307–2359, 2010.

[PPV11] Yury Polyanskiy, H. Vincent Poor, and Sergio Verdu. "Feedback in the Non-Asymptotic Regime." *IEEE Transactions on Information Theory*, **57**(8):4903–4925, 2011.

[Ric94] M. Rice. "Comparative analysis of two realizations for hybrid-ARQ error control." In *1994 IEEE GLOBECOM. Commun.: Communications Theory Mini-Conference Record,*, pp. 115–119, 1994.

[Sah08] A. Sahai. "Why Do Block Length and Delay Behave Differently if Feedback Is Present?" *IEEE Trans. Inf. Theory*, **54**(5):1860–1886, 2008.

[Sch71] J. Schalkwijk. "A class of simple and optimal strategies for block coding on the binary symmetric channel with noiseless feedback." *IEEE Transactions on Information Theory*, **17**(3):283–287, May 1971.

[Sch96] L. J. Schulman. "Coding for interactive communication." *IEEE Transactions on Information Theory*, **42**(6):1745–1756, 1996.

[SF11] O. Shayevitz and M. Feder. "Optimal Feedback Communication Via Posterior Matching." *IEEE Transactions on Information Theory*, **57**(3):1186–1222, March 2011.

[SF16] O. Shayevitz and M. Feder. "A Simple Proof for the Optimality of Randomized Posterior Matching." *IEEE Transactions on Information Theory*, **62**(6):3410–3418, June 2016.

[SH16] R. T. Sukhavasi and B. Hassibi. "Linear Time-Invariant Anytime Codes for Control Over Noisy Channels." *IEEE Transactions on Automatic Control*, **61**(12):3826–3841, 2016.

[Sha48] C. E. Shannon. "A mathematical theory of communication." *The Bell System Technical Journal*, **27**(3):379–423, 1948.

[SP73] J. Schalkwijk and K. Post. "On the error probability for a class of binary recursive feedback strategies." *IEEE Transactions on Information Theory*, **19**(4):498–511, July 1973.

[SPK18] Oron Sabag, Haim H. Permuter, and Navin Kashyap. "Feedback Capacity and Coding for the BIBO Channel With a No-Repeated-Ones Input Constraint." *IEEE Tran. Inf. Theory*, **64**(7):4940–4961, 2018.

[Tru14] Lan V. Truong. "Posterior matching scheme for Gaussian multiple access channel with feedback." In *2014 IEEE Information Theory Workshop (ITW 2014)*, pp. 476–480, 2014.

[TT02] A. Tchamkerten and E. Telatar. "A feedback strategy for binary symmetric channels." In *Proc. IEEE Int. Symp. Inf. Theory*, pp. 362–362, June 2002.

[TT06] A. Tchamkerten and I. E. Telatar. "Variable length coding over an unknown channel." *IEEE Transactions on Information Theory*, **52**(5):2126–2145, May 2006.

[VRD16] Kasra Vakilinia, Sudarsan V. S. Ranganathan, Dariush Divsalar, and Richard D. Wesel. "Optimizing Transmission Lengths for Limited Feedback With Nonbinary LDPC Examples." *IEEE Transactions on Communications*, **64**(6):2245–2257, 2016.

[Wil14] Adam Royce Williamson. *Reliability-output Decoding and Low-latency Variable-length Coding Schemes for Communication with Feedback*. PhD thesis, University of California, Los Angeles, 2014.

[WWB18] R. D. Wesel, N. Wong, A. Baldauf, A. Belhouchat, A. Heidarzadeh, and J.-F. Chamberland. "Transmission Lengths That Maximize Throughput of Variable-Length Coding & ACK/NACK Feedback." In *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2018.

[YKE21] Recep Can Yavas, Victoria Kostina, and Michelle Effros. "Variable-length Feedback Codes with Several Decoding Times for the Gaussian Channel." In *2021 IEEE Intl. Symp. Inf. Theory (ISIT)*, pp. 1883–1888, 2021.

[YKE24] Recep Can Yavas, Victoria Kostina, and Michelle Effros. "Variable-Length Sparse Feedback Codes for Point-to-Point, Multiple Access, and Random Access Channels." *IEEE Tran. Inf. Theory*, **70**(4):2367–2394, 2024.

[YPA21] Hengjie Yang, Minghao Pan, Amaael Antonini, and Richard D. Wesel. "Sequential Transmission Over Binary Asymmetric Channels With Feedback." *IEEE Tran. Inf. Theory*, 2021.

[YW19] Hengjie Yang and Richard D. Wesel. "Finite-Blocklength Performance of Sequential Transmission over BSC with Noiseless Feedback." In *2021 IEEE International Symposium on Information Theory (ISIT)*, 2019.

[YYK22] Hengjie Yang, Recep Can Yavas, Victoria Kostina, and Richard D. Wesel. "Variable-Length Stop-Feedback Codes With Finite Optimal Decoding Times for BI-AWGN Channels." In *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 2327–2332, 2022.