Anytime Information Theory

by

Anant Sahai

B.S., University of California at Berkeley (1994)
S.M., Massachusetts Institute of Technology (1996)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2001

Author . . . .
Department of Electrical Engineering and Computer Science
November 30, 2000

Certified by . . . .
Sanjoy K. Mitter
Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Anytime Information Theory
by
Anant Sahai

Submitted to the Department of Electrical Engineering and Computer Science
on November 30, 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

We study the reliable communication of delay-sensitive bit streams through noisy channels. To bring the issues into sharp focus, we focus on the specific problem of communicating the values of an unstable real-valued discrete-time Markov random process through a finite-capacity noisy channel so as to have finite average squared error from end-to-end. On the source side, we give a coding theorem for such unstable processes that shows that we can achieve the rate-distortion bound even in the infinite horizon case if we are willing to tolerate bounded delays in encoding and decoding. On the channel side, we define a new parametric notion of capacity called anytime capacity. The corresponding sense of reliable transmission is stronger than the traditional Shannon capacity sense but is less demanding than the sense underlying zero-error capacity. We show that anytime capacity exists for memoryless channels without feedback and is connected to standard random coding error exponents. The main result of the thesis is a new source/channel separation theorem that encompasses unstable processes and establishes that the stronger notion of anytime capacity is required to be able to deal with delay-sensitive bit streams. This theorem is applied in the control systems context to show that anytime capacity is required to evaluate channels if we intend to use them as part of a feedback link from sensing to actuation. Finally, the theorem is used to shed light on the concept of "quality of service requirements" by examining a toy mathematical example for which we prove the fundamental necessity of differentiated service without appealing to human preferences.

Thesis Supervisor: Sanjoy K. Mitter
Title: Professor

Acknowledgments

I thank my advisor Professor Sanjoy Mitter for being such an inspiring role model as a researcher. Not only did his ideas and insights help shape this work, but watching his aesthetic sense and intuition in action taught me what good research is truly about. He was also very supportive and patient with the time and twists that this research program has taken. I would also like to thank Professors Robert Gallager, John Tsitsiklis, and Dave Forney for their helpful comments and guidance as members of my committee. I also appreciate the many interactions I had with Professor Vivek Borkar. It was some of our work with him that helped prompt me to revisit the whole idea of capacity and take a fresh look. More recently, the discussions I have had with Professors Venkat Anantharam, Shlomo Shamai, and Emre Telatar have helped me present the ideas in this thesis in a clearer way.

I thank all my fellow students at LIDS and MIT who helped make my time here very enlightening. In particular, I valued my close collaborations with my office-mate Sekhar Tatikonda who not only served as a useful sounding board for ideas, but also was ever helpful with references and pointers into the literature. S.R. Venkatesh has also been a friendly source of different perspectives and interesting discussions. The many people I met through the student organizations Sangam and Asha helped broaden my non-academic interests. The administrative staff of LIDS, the EECS department, and MIT have also been a great help in navigating the system.

Last but not least, I want to thank my brother, parents, and the Gods for their constant love and support.

This research was partially supported by a DOD/ONR fellowship from the U.S. Government and by grants ECS-9873451 from NSF and contracts DAAD19-00-1-0466 and DAAL03-92-G-0115 from the Army Research Office. The flexibility and academic freedom made possible by these sponsors are much appreciated.

Contents

List of Figures

Chapter 1

Introduction

"A bit is a bit is a bit" seems to have become the motto of this digital age. Though we often take it for granted, the existence of such a common currency for information is far from obvious. While in practice it has been justified by the rapid advancement of digital electronics and the rise of a communications infrastructure to match, its meaningfulness rests upon the fundamental theorems of information theory — the philosophical foundation that helps us gain insight into problems of communication.

Broadly speaking, there are two sides to information theory. The better known side deals with the transmission of bits over channels. The noisy channel coding theorems establish the fundamental limitations for reliable transmission of bits over noisy channels. But there is another side to information theory. It relates to the encoding of sources with respect to some criterion of fidelity. The rate distortion theorems establish fundamental tradeoffs between fidelity and the length of the encoding in bits. These two sides of information theory are joined together by the information transmission theorems relating to source/channel separation. Philosophically speaking, these theorems establish that the notion of "reliable transmission" used for discussing channels is compatible with the notion of "bits" used for encoding sources.

Of course, information theory has come a long way since Shannon's seminal papers.[60, 62] On the channel side, there has been much work over the years in extending this theory to cover broader classes of channels[67], by now covering very useful approximations to unreliable media that are encountered in practice, including many situations with memory. Though less dramatic, the source side has also advanced considerably and now covers many general classes of sources with memory.[37, 7] The limiting factor on source coding has traditionally been a matter of finding good theoretical approximations to real world situations. The information transmission theorems have also been extended along the way[6, 66], even partially into contexts of networked collaborative computing.[58, 52]

However, there are still many issues in modern communication systems for which information theoretic understanding eludes us. Networking in particular has a whole host of them, leading Ephremides and Hajek to entitle their recent survey article "Information Theory and Communication Networks: An Unconsummated Union!" [15]. They comment:

> The interaction of source coding with network-induced delay cuts across the classical network layers and has to be better understood. The interplay between the distortion of the source output and the delay distortion induced on the queue that this source output feeds into may hold the secret of a deeper connection between information theory. Again, feedback and delay considerations are important.

These issues center around “information streams” where the data source continues to generate information forever rather than being a single isolated message. In these contexts, people have noticed that streams emerging from different real-world sources seem to behave differently. What is appropriate for a classic file transfer may not be suitable for real-time multimedia signals.

Simultaneously, an important class of mathematical sources has eluded complete understanding in an information theoretic sense: unstable sources. By these, we are referring to certain non-stationary processes which often arise in control and other system-theoretic modeling contexts. Although they have proven themselves to be valuable parts of the system modeler's toolbox, the fundamental difficulty with unstable processes is that they tend to grow with time and they have asymptotically infinite variance. Because of this, their “streaming” character through time cannot be ignored. We believe that extending information theory to address unstable sources might be the conceptual key to understanding the deeper issue of “streaming” in general.

In control contexts, the objective is to keep the modeled system stable. To do this, communicating the values of the underlying unstable process from one place to another is an important issue[69, 70, 54] since the physical location for applying actions is generally somewhat removed from the location where observations are being made. In practice, people are already using quantizers and digital implementations to control unstable physical systems and there has been some isolated theoretical work on trying to extend source coding theory to unstable processes. Most recently, the work of Sekhar Tatikonda [63] has taken us significantly forward in understanding the information theoretic limitations of control systems. Even so, a major gap remains. Till now, there have been no information transmission theorems to tie the source coding of unstable processes to communication over noisy channels.

This question has remained open for quite some time. To make matters concrete, we start with a very simple example of a random walk on the infinite line with known initial condition. We then show how even with a noiseless channel, a block source code cannot be used to track this unstable process. All block source codes have asymptotically infinite distortion, regardless of the rate or how large a block length is chosen. For our particular example however, there is an obvious causal source encoder with memory which tracks the source perfectly over a noiseless link. To illustrate the issues that arise with noisy links without feedback, we show how traditional “linear” approaches to estimation fail because the effective signal-to-noise ratio goes to zero with time. The point of this section is to show that even in this simple case, the problem is far from trivial.

After these introductory comments, we introduce the general problem of tracking unstable scalar Markov processes over noisy channels, ask the relevant questions, and informally state the major thrusts and results of this thesis.

In the second chapter of this thesis, we review definitions of sources, channels, codes, etc. After reviewing channel coding in a streaming context, we illustrate how any attempt to cascade our simple source encoder with a conventional channel encoder is doomed to failure because the Shannon sense of reliable transmission is not sufficient for our task. This will show that in the context of unstable processes, a “bit” is not necessarily a “bit” and we will make this idea much more precise in the next chapter by introducing a delay-dependent notion of intrinsic meaning for streams of bits that shows how bit-streams can be fundamentally different from each other even if they have the same bit-rate.

Next, we introduce a new parametric notion of reliable transmission and its associated channel capacity that we call anytime capacity and suggest how anytime decoders can be used to avoid the problem for our simple random walk. Unlike classical decoders, an anytime decoder will eventually correctly recover the values for *all* the bits originally sent. We then use a random coding argument to show that the anytime capacity of a binary erasure channel and the power-constrained AWGN channel is non-zero even without any feedback but implicitly assuming access to common randomness shared by the encoder and decoder. After that motivation, we quickly establish the connection between anytime capacity and general block random coding error exponents. We also show the existence of deterministic codes with the required properties.

Finally, we show that the random walk can be tracked using an anytime channel code to transport the bit-stream and give the main result of this thesis: the relevant information transmission theorem for unstable scalar Markov processes. This information transmission theorem establishes that the unstable processes are fundamentally different from memoryless or stationary processes. The traditional source-channel separation theorems imply that the bit-streams used to encode all memoryless or stationary processes are qualitatively alike as long as we are willing to tolerate delays. All they need is a channel with sufficient capacity. But even if we are willing to live with delay, unstable processes require more from a channel than merely Shannon capacity.

This understanding is extended to simple control problems and then used to establish the anytime capacities of the erasure channel with feedback and the AWGN channel with feedback. In the erasure case, we are able to get a parametric closed-form expression for the true anytime capacity. For the AWGN case, we are able to recover a result reminiscent of Kailath and Schalkwijk[59] which gives a doubly-exponential in delay convergence of errors to zero. The difference is that our scheme requires no blocking of the bits and works for all delays at once! The same technique lets us get a lower bound on the zero-error capacity of power-constrained additive noise channels where the channel noise has bounded support.

In a sense, the parameter in our definition of anytime capacity can be interpreted as an additional Quality of Service requirement. We illustrate that in the penultimate chapter by giving a simple example where the need for differentiated service can be established mathematically. To our knowledge, this is the first time where a simple class of random processes have been exhibited which intrinsically require different Qualities of Service apart from just bit-rate. This requirement is not posited *a-priori* or by appeal to human preferences, but emerges from the nature of the source and distortion measure.

1.1 "Streaming" and Why We Study Unstable Processes

At one level, unstable processes are interesting in their own right since they emerge naturally in the course of system modeling. Addressing them from an information theoretic perspective is useful to other fields like control which might have communication problems embedded in their contexts. But at a deeper level, we are interested in them because they represent a conceptually simple and yet extreme case. For unstable processes, the issues of streaming, delays, and "real-time" are absolutely essential to the problem of end-to-end communication. For us, the non-stationarity of unstable processes is a way of forcing attention on their streaming aspects since we show that the standard tricks of dividing the source stream into a sequence of independent message blocks cannot work.

In the literature, there are other information theoretic approaches that shed some light on the general issue of "streaming." Here we will briefly mention two of them: Tse's approach of adjusting the rate of source coding in response to buffer states and Shulman's approach to interactive computation over noisy channels.

1.1.1 Adjustable Length Source Codes: Tse

Tse's dissertation [65] shows how adjusting the coding of streaming sources in response to the state of the communication system that they are feeding into can give much improved performance. There are fewer buffer overflows while we still maintain good end-to-end distortion. Since the delays in the model are due to buffering, this strategy also reduces the end-to-end delays, even when the buffer sizes are infinite.

From our perspective, Tse's approach represents an important step towards understanding "streaming" when the acceptable end-to-end delay is thought of as being smaller than the end-to-end delay that would be introduced by a fixed-rate source coding system connected to the same communication system. If we restrict ourselves to the narrow classical information-theoretic perspective that we do not care about end-to-end delay as long as it is finite, Tse's approach does not buy us anything in terms of either average rate or distortion. But by introducing a loose coupling between the source and channel coding, it does give us a much better tradeoff between delay and performance than the classical completely separated approach.

1.1.2 Interactive Computation: Schulman

Schulman's work [58] addresses the problem of how to interactively compute using distributed processors if the communication links connecting the processors are noisy channels. The issue of "streaming" is central here since the messages to be sent from one processor to the other depend on the messages already received from the other processor. Grouping the messages into blocks for traditional block channel-encoding is out of the question since there is no way to know what messages we will want to send in the future without already sending and receiving messages now. He shows that for any sufficiently large problem size, there exist good coding strategies that allow the computation to proceed with an arbitrarily low probability of error.

As such, the essential "streaming" aspect is understood to emerge from the interaction between the two streams of messages. The results are conceived in terms of a single finite-sized problem instance rather than infinite duration streams. This line of work has effectively given new information transmission theorems that cover the case of interactive computation and have been extended by others [52]. In the spirit of computer science, one of the goals has been to make sure that the complexity does not grow by more than a polynomial factor. In contrast, our work on unstable processes in this thesis covers a problem with a single information stream in a standard one-way communication problem. However, it still shares a lot with Schulman's approach in spirit and technique.

1.2 The Simplest Case: Tracking a Simple Random Walk

Random walks are among the simplest examples of unstable processes. To make matters concrete, let us consider the following extremely simple discrete time real valued source X_t which is generated as follows: $X_0 = 0$ and $X_{t+1} = X_t + W_t$ where the $\{W_t\}$ are i.i.d. fair coin tosses with $P(W_t = -1) = P(W_t = 1) = \frac{1}{2}$. It should be clear that $\{X_t\}$ is not stationary and can wander over all the integers (a countable set) in its random walk. In fact, it turns out that while $E[X_t] = 0$ for all t, $E[X_t^2] = t$ and thus tends to infinity by construction.

To talk about tracking this random walk, we need to have a per-letter distortion measure. We will use the usual squared-error distortion $\rho(X, \hat{X}) = (\hat{X} - X)^2$. The following definition specifies what we mean by "tracking."

Definition 1.2.1 *A random process $\{\hat{X}_t\}$ is said to* ρ-track *another process $\{X_t\}$ if $\exists D < \infty$ such that $\sup_{t>0} E[\rho(X_t, \hat{X}_t)] = D < \infty$.*

We are interested in the problem of tracking this random walk through a "finite capacity" link. We will first consider noiseless finite rate binary channels and then consider a noisy real valued channel to show some of the different issues that can arise.

1.2.1 Block-codes are not enough

A great deal of attention in coding and information theory has been dedicated to block-codes and analyzing their properties.[23] In order to show their shortcomings, we need to know what we mean by a "block-code" for sources.

Definition 1.2.2 *A* block source-code *with block-length n and rate R (where both n and nR are assumed to be non-negative finite integers) is a pair of functions (F_n, G_n) such that F_n maps a vector $(X_{in}, X_{in+1}, \ldots, X_{in+n-1})$ (representing n consecutive symbols from the source sequence) into a binary string s of length nR (thought of as a non-negative integer less than 2^{nR}) and G_n maps binary strings s of length nR into a reconstruction vector $(\hat{X}_{in}, \hat{X}_{in+1}, \ldots, \hat{X}_{in+n-1})$. We call F_n the* block source-encoder *and G_n the* block source-decoder.

This is slightly different from some standard definitions of a block source-code which do not explicitly include the decoder and instead use a single function going directly from a block of source samples to a block of reconstructions. We use our definition because it makes it easier to see what it means to interconnect a source-code with a channel-code since both explicitly use the common currency of binary bits.

Now, we are ready to state and prove the initial negative result.

Proposition 1.2.1 *No block source-code can track the random walk $\{X_t\}$, regardless of the block-length n or the rate R.*

Proof: Let (F_n^R, G_n^R) be a block-code of rate R and block-length n. Let the maximum size element of the reconstruction vector given input s be denoted by $M(s) = \max_{j<n} \|G_n^R(s)_j\|$. Since by definition the domain of G_n^R is the finite set $\{0, 1, \ldots, 2^{nR} - 1\}$, we know that there exists a finite integer M such that $M(s) < M$. This means that for all t, $\|\hat{X}_t\| < M$.

But the random walk X_t will eventually spend arbitrarily large amounts of time outside any bound. Since X_t is the sum of t i.i.d. random variables with zero mean and unit variance, by a simple application of the central limit theorem we know $\forall T > 0$ there exists $N_T > 0$ such that $\forall t > N_T$, we have $P(|X_t| > \sqrt{t} > T) > \frac{1}{4}$. So, $\limsup_{t\to\infty} E[\rho(X_t, \hat{X}_t)] > \frac{1}{4}(T - M)^2$. But T was arbitrary and so we have $\limsup_{t\to\infty} E[\rho(X_t, \hat{X}_t)] = \infty$ for all block-codes, thereby proving the theorem. □

Notice that the proof only relied upon the fact that the *source decoder* was a deterministic function on a finite set and hence had a finite range. Thus, we can immediately extend this result to the case of block joint-source-channel-codes in those cases where the channel has a finite output alphabet giving us the following Corollary, which reduces to the previous theorem in the case of a noiseless channel.

Corollary 1.2.1 *If the communication channel has a finite output alphabet, no block joint-source-channel-code can track the random walk $\{X_t\}$, regardless of the block-length n.*

This rejection of block codes when dealing with unstable processes tells us that the arguments of Krich [40, 41] on sources with delay-dependent fidelity criteria are not applicable directly since he restricts his attention to block-codes throughout. We discuss this more in Section 2.2.4.

1.2.2 A Source Code with Memory

The negative results for block codes are interesting because we intuitively know that this source should be able to be perfectly tracked with a rate 1 causal encoder that simply encodes $W_t = X_{t+1} - X_t$ with a single bit corresponding to its sign. Mathematically, we have:

$$S_t = F_t(X_1^t) = \frac{(X_t - X_{t-1} + 1)}{2} \tag{1.1}$$

As we can see, the encoder only needs access to the current and previous value of the source, not the entire history.

All the decoder has to do is to add up the W_t values up till now. So:

$$\hat{X}_t = G_t(S_1^t) = \sum_{i=1}^{t}(2S_i - 1) \tag{1.2}$$

which can also be done in a a simple recursive manner using an internal integer variable to track $\hat{X}_{t-1}$. In this example, the reconstruction clearly tracks the original perfectly, since $\hat{X}_t = G_t(F_1^t(X_1^t)) = \sum_{i=1}^{t}(2\frac{(X_i - X_{i-1}+1)}{2} - 1) = \sum_{i=1}^{t}(X_i - X_{i-1}) = X_t$. Of course, this is all *assuming that the input to the decoder is exactly the output of the encoder.*

1.2.3 Tracking across Noisy Channels

The simple random walk example is clearly a well behaved linear system on the real line. Our experience with filtering for linear systems[36] might suggest that it is easy to track given linear observations with additive noise. If we want to use a linear encoder with Gaussian noise, our observation equation will be of the form: $Y_t = c_t X_t + V_t$ where $c_t X_t$ is our linear encoding of the signal and $\{V_t\}$ is a white Gaussian process with zero mean and unit variance.

But things are not so simple since real communication channels have constraints on their inputs. The additive white Gaussian noise channel (defined formally in section A.4.1) is traditionally considered with a power constraint P on the average power in the channel input signal. It turns out that if our encoder has access to noiseless feedback then the problem is not that difficult. Our analysis in [55] can be used to show that a simple simulated control system can be used to track the random walk process while still meeting the power constraint. In that case, we do not transmit $c_t X_t$ but rather $c_t u_t$ where $\{u_t\}$ is the simulated control signal used to keep the simulated plant stable. This particular scheme is discussed further in Section 7.2.2.

The case without access to feedback is far more tricky. Without feedback, there is no straightforward way to have a simulated control system and so we have to transmit functions of X_t itself. Since $E[X_t^2] = t$, it is clear that no constant c_t will meet the finite power constraint. We have to use a time-varying encoder with $c_t \leq \frac{\sqrt{P}}{\sqrt{t}}$ in order to meet it.

We can then generate estimates using the corresponding time-varying Kalman filter.[36] Since we know that the estimates can only get better if c_t is larger, it suffices to look at the case where c_t is equal to its upper bound. If we explicitly write out the recursive expression for the Kalman filter error covariance, we get:

$$\begin{aligned}\sigma_{t+1}^2 &= \frac{\sigma_t^2 + 1}{\frac{P}{t+1}(\sigma_t^2 + 1) + 1} \\ &= \frac{1}{\frac{P}{t+1} + \frac{1}{\sigma_t^2+1}}\end{aligned}$$

with an initial condition of $\sigma_0^2 = 0$ since we know exactly where the random walk starts. If we evaluate this recursion numerically on a computer, we see that the variance grows as $O(\sqrt{t})$ with time t.

We can also see this more rigorously. As a function of t, the estimation variance σ_t^2 can either get arbitrarily large or stay bounded. To show that this variance has no finite upper bound, we first show that σ_t^2 increases with time as long as $\sigma_t^2 < -\frac{1}{2} + \sqrt{\frac{t+1}{P} + \frac{1}{4}}$ by noticing that:

$$\begin{aligned}1 &< \frac{\sigma_{t+1}^2}{\sigma_t^2} \\ 1 &< \frac{1}{\frac{\sigma_t^2 P}{t+1} + \frac{\sigma_t^2}{\sigma_t^2+1}} \\ \frac{\sigma_t^2 P}{t+1} &< \frac{1}{\sigma_t^2 + 1} \\ (\sigma_t^2)^2 P + \sigma_t^2 P - t - 1 &< 0 \\ \sigma_t^2 &< -\frac{1}{2} + \sqrt{\frac{t+1}{P} + \frac{1}{4}}\end{aligned}$$

are equivalent statements if we know that σ_t^2 must be positive. If we assume that σ_t^2 stays bounded, we know that for large t it must be strictly increasing up to a some threshold M.

To now see that it must exceed all thresholds M, we just consider a time $t > 2(M + \frac{3}{4})^2 P - 1$ and consider what happens if it is within $\frac{1}{4}$ of the threshold M at time t. Then:

$$\begin{aligned}\sigma_{t+1}^2 &= \frac{\sigma_t^2 + 1}{\frac{P}{t+1}(\sigma_t^2 + 1) + 1} \\ &> (M - \frac{1}{4} + 1)(1 - \frac{\frac{(\sigma_t^2+1)P}{t+1}}{\frac{(\sigma_t^2+1)P}{t+1} + 1} \\ &> (M + \frac{3}{4})(1 - \frac{\frac{(M+\frac{3}{4})P}{t+1}}{\frac{(\sigma_t^2+1)P}{t+1} + 1} \\ &> (M + \frac{3}{4})(1 - \frac{(M + \frac{3}{4})P}{t+1}) \\ &= M + \frac{3}{4} - \frac{(M + \frac{3}{4})^2 P}{t+1}\end{aligned}$$

$$
\begin{aligned}
&> M + \frac{3}{4} - \frac{(M+\frac{3}{4})^2 P}{(2(M+\frac{3}{4})^2 P - 1) + 1} \\
&= M + \frac{1}{4}
\end{aligned}
$$

showing that it crosses the threshold M. But M was arbitrary and so we can take an increasing sequence starting at 0 and going up by $\frac{1}{4}$ each to show that all thresholds are eventually crossed regardless of the power constraint P. This proves the following theorem:

Theorem 1.2.1 *For all power constraints $P > 0$, no time-varying linear memoryless encoder can be used to track our simple random walk across an additive white Gaussian noise channel with power constraint P without access to feedback.*

Other methods based on encoding the innovation signal directly suffer the problem of having to keep the encoder and decoder perfectly "in sync." We will show this more precisely in Section 2.4.2. This difficulty has long been recognized. In his definitive book on rate-distortion theory, Toby Berger writes:([6] pg 247)

> It is worth stressing that we have proved only a source coding theorem for the Wiener process, not an information transmission theorem. If uncorrected channel errors were to occur, even in extremely rare instances, the user would eventually lose track of the Wiener process completely. It appears (*although it has never been proved* [emphasis mine]) that, even if a *noisy* [emphasis in original] feedback link were provided, it still would not be possible to achieve a finite MSE per letter as $t \to \infty$.

One of the goals of this thesis is to disprove this longstanding conjecture.

1.3 General Problem Statement

The above discussion of the simple random walk was just intended to motivate the general issues and to show that they are far from trivial. In general, we are interested in the problem of communicating a Markov process across a noisy channel:

Given a scalar discrete-time Markov source $\{X_t\}$ with parameter a and driven by noise $\{W_t\}$:

$$X_{t+1} = AX_t + W_t$$

and a discrete-time $(0, \tau, \mathcal{A}, \mathcal{B})$ noisy channel with 0 delay and sampling time τ, that maps inputs from $\mathcal{A}$ randomly into outputs in $\mathcal{B}$, is it possible to design encoders and decoders within a specified finite end-to-end delay constraint so that the output of the decoder $\{\hat{X}_t\}$ achieves a desired mean-squared performance $\sup_{t>0} E[(\hat{X}_t - X_t)^2] = D$?

As illustrated in Figure 1-1, this problem has a few important features. The first is that the source sequence $\{X_t\}_{t=1}^{\infty}$ is not realized all at once, but is observed in "real time" as time progresses. In the same way, the channel is also only available to be used once every τ units of time. As a result, it only makes sense that the reconstructed estimates $\{\hat{X}_t\}_{t=1}^{\infty}$ are also generated one at a time in a somewhat "real time" fashion. By finite end-to-end delay constraint d, we mean that the estimate $\hat{X}_t$ for X_t must be ready at the decoder within d time units — that is, it must be ready before time $t + d$. In other words, all the blocks in the above diagram can and should be viewed as being causal in the appropriate sense. Their outputs are not allowed to depend on random variables that have not occurred yet. The following questions arise naturally and are answered in this Thesis:

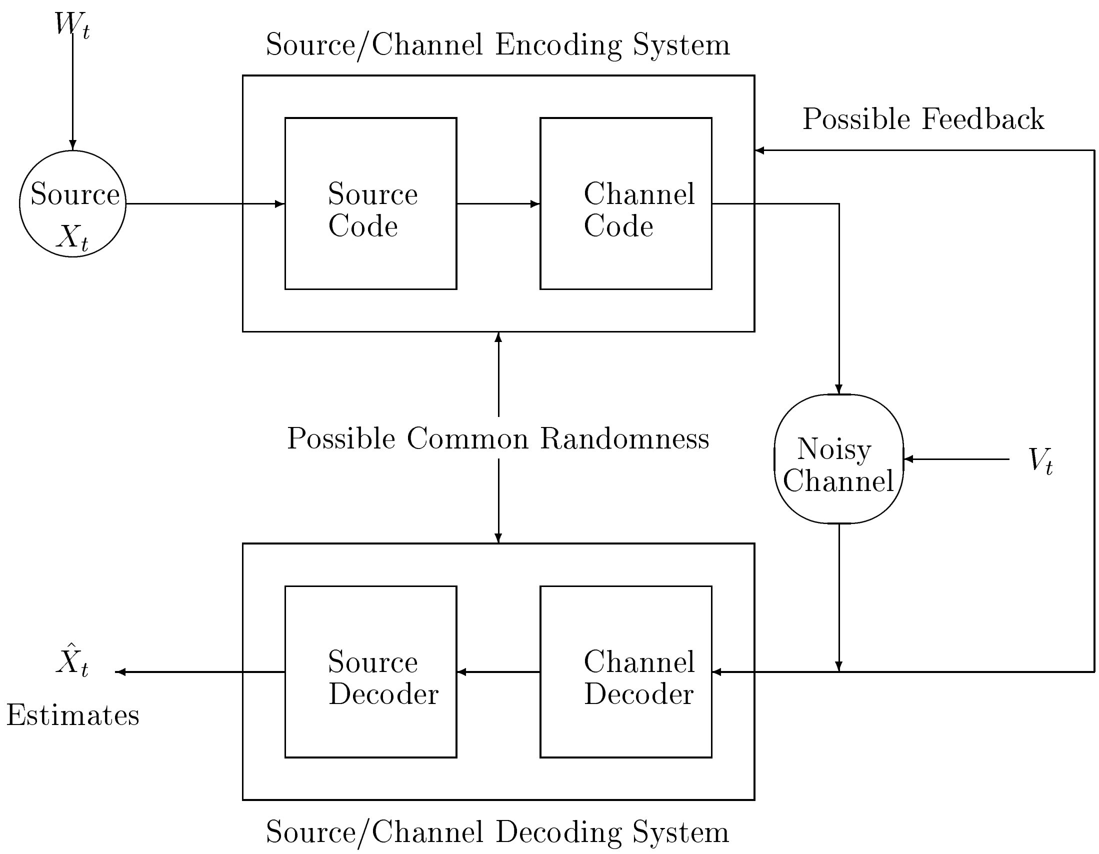

Figure 1-1: The estimation problem across a noisy channel

- For a finite end-to-end delay constraint, can we ever get D finite if the process is unstable (ie $A \geq 1$)?

- Is it possible to accomplish this task using encoders which have access to noiseless feedback of the channel outputs?

- Is it possible to accomplish this task using encoders which do not have access to any feedback?

- Is there a necessary and sufficient condition on the channel that must hold for finite D to exist even if we impose no restrictions on the joint source/channel encoding and decoding systems?

- Can we express this condition in terms of reliably communicating "bits" in some sense?

- Is it possible to somehow factor solutions through "bits" — giving the joint encoder and decoder systems in terms of separate source and channel codes?

- What do the encoder and decoder need? Do they have to be random and have access to common randomness?

- Can we bound the performance and the optimal D if we allow ourselves to consider the limit of increasing end-to-end delays as we do in existing rate-distortion theory?

- Is this bound achievable over a wide class of suitable noisy channels?

- Can we extend these results to performance measures other than the expected second moment of the error?

- What are the implications for Control, "Quality of Service," and more general problems?

1.4 Main Results

After introductory and review material, the thesis has four main theoretical thrusts. Here we informally state the main result of each. While the source coding and channel coding discussions are intelligible even in isolation, the main contribution of the thesis is to give a unified picture that lets us look at the whole problem and to see how the pieces fit together in the spirit of Shannon's original program for understanding communication. The idea is to look at extremely simple situations and use them to develop insights that shed light on certain core problems of communication.

In the final chapters of this thesis, we use the seemingly simple unstable scalar Markov sources and the results described below to illuminate issues of "real-time," "streaming sources," "control over communication links," and "differentiated quality of service." Without looking at the whole picture, it is impossible to understand the real issues and interconnections underlying such problems.

1.4.1 Source Coding

Lossy source coding is about translating the original source process into an approximate version of that process by first going through an intermediate process expressed in the language of bits. The intermediate process consists of a stream of bits generated based on the realization of the original source. It is subsequently mapped into the reconstruction alphabet to get the approximate version. We can naturally speak of the bit-rate for the intermediate stream, while distortion is measured between the original process and the final approximating version. In Chapter 4, a source coding theorem is proved for unstable Markov processes. The theorem works for the infinite-horizon average distortion case and thereby resolves a problem left open since [29].

Result 1.4.1 *For the unstable scalar Markov source, there exist variable rate source codes, all with finite end-to-end delay, which can approach the fundamental rate-distortion bounds in both rate and distortion.*

The important feature of this result is that we are dealing with a streaming context. Unlike the standard setup [30], we cannot assume that the source realization is known in advance at time 0. It is only observed as time goes on. Furthermore our approach works even though the source is severely non-stationary and exhibits strong dependencies that never fade away between any two blocks of time.

1.4.2 Delay Sensitivity

In Chapter 3, a formal definition is given for the delay and error dependent "complexity" of source codes and thus also of the bit-streams that emerge from them. This definition focuses on the sensitivity of a code to the propagation of errors. It results in a sense of intrinsic complexity for a pair consisting of a random source together with the distortion measure. In some ways, this parallels the way that the theory of computational complexity is able to use a complexity measure for algorithms to determine the intrinsic complexity for problems. This error-propagation complexity is explored for the Markov sources and is used to establish:

Result 1.4.2 *For the unstable scalar Markov source, all source codes which track the process in the mean-squared sense are sensitive to bit-errors. If any of the bits for the last d time units have been altered, the resulting additional distortion can grow exponentially as A^{2d}.*

This sort of fundamental sensitivity to errors tells us that standard approaches to cascading source and channel codes can never be successful because even a single uncorrected error, no matter how rare, can eventually cause unboundedly large distortions. We call bitstreams "weak" if they result from codes that have this cascading-error property. In contrast, bitstreams coming from traditional codes where errors cannot cascade catastrophically are called "strong."

1.4.3 Anytime Channel Coding

Channel coding is about reliable transporting a stream of bits across a noisy channel. Recognizing that traditional conceptions of reliable transmission are inadequate for "weak" bitstreams, a definition is given for a new stronger sense of reliable transmission and its associated capacity measure that we call anytime capacity. Rather than thinking of "reliable transmission" as being able to achieve a specified low probability of error, we think of the probability of error as being determined only when the user of the information has committed to a particular delay or wait. The notion of reliability can be parametrized by a function which tends to zero with increasing delay. Even if the probability never actually reaches 0 for a finite wait, if it goes down fast enough we can show that eventually every bit will be decoded correctly. This sense is in some ways the natural generalization of the sense in which TCP/IP achieves reliable transport of bit streams across a noisy Internet.

Result 1.4.3 *For discrete memoryless channels without feedback, encoders exist for which the decoder has the freedom to choose the delay of decoding and so that the probability of error for any bit position tends to zero exponentially with increasing delay.*

We study this anytime capacity for particular channels using random coding arguments and show it to be non-zero. In the general case, we relate it to standard block random coding error exponents and show how this sense of reliable transmission can be interpreted as a "universal" error exponent, where the universality of the encoder is over the delay that is acceptable to the decoder. This is related to work showing that the random coding error exponents govern the rate at which the probability of error for finite state convolutional codes goes down if we force decoding to occur with a fixed delay rather than letting Viterbi decoding take its course.[19]

1.4.4 Separation Theorem

In chapter 6, a new information transmission theorem is developed that uses the notions developed so far and ties them together to give the relevant source/channel separation for unstable Markov processes.

Result 1.4.4 *For the unstable scalar Markov source with parameter $A > 1$ to be tracked in the finite mean-squared error sense across a noisy channel by some encoding/decoding system, it is necessary for the channel to be able to carry at least $\log_2 A$ bits per unit time with a probability of error that tends to zero exponentially as $2^{-(2\log_2 A)d}$ with delay $d > 0$ chosen at the decoder. Furthermore, this condition is also sufficient and tracking can be accomplished with a system that is divided into a source code (which outputs bits) and a channel code which focuses on getting the bits across the channel.*

This establishes that the sense of reliable transmission that we have defined is fundamental for this problem. The key to this result is the proof of the converse theorem. While for traditional Shannon capacity and rate-distortion theory the converse is a simple consequence of the properties of mutual information, in our case the converse is much more difficult to prove since we do not have a mutual-information characterization of anytime capacity. We give a constructive proof of the converse that is able to relate the two operational notions to each other without having to go through another characterization.

By rigorously giving us something other than just the bit-rate to match between source code and channel code, this theorem is a significant step. It brings us several steps closer towards a rigorous understanding of prior ideas, motivated by intuition, of loosely coupled joint source/channel coding like those in [32].

Chapter 2

Background

2.1 Sources, Channels, and Codes

Here we quickly review some basic concepts (covered more precisely in Appendix A) for those unfamiliar with the basic information theoretic formulation of communication problems.

2.1.1 Sources

For our purposes, a source is a random process $\{X_t\}$ evolving in discrete time ($t \in \{0, 1, 2, \ldots\}$) over some alphabet $\mathcal{X}$. Such sources are used as approximations to the real world data generating processes that we are interested in. Our main focus is on scalar valued linear Markov processes.

Definition 2.1.1 *Given a real number A, and real valued i.i.d. random variables $\{W_t\}$ the* scalar discrete-time Markov source with parameter A and noise W *is defined by:*

$$\begin{aligned} X_0 &= 0 \\ X_t &= AX_{t-1} + W_t \end{aligned}$$

This can be expressed without recursion as:

$$X_t = \sum_{i=1}^{t} A^{t-i} W_t$$

There are four distinct regions of the parameter A. If $A = 0$, then the source is white. If $0 < |A| < 1$, then the source is stable since even as $t \to \infty$, the random variables X_t stay well behaved. In particular, if W_t has zero-mean and variance σ^2, then X_t asymptotically has zero-mean and variance $\frac{\sigma^2}{1-A^2} < \infty$.

The other two cases are not so well behaved. If $|A| = 1$, then $\{X_t\}$ is a random walk and is unstable since the uncertainty grows unboundedly in time. It should be clear how our simple random walk falls into this category. Finally if $|A| > 1$, the process is exponentially unstable.

2.1.2 Source Codes

The idea of a source code is to translate a source process into a process taking values in the alphabet of binary strings. It is expressed in two parts: the encoder and the decoder. The encoder maps from the source alphabet $\mathcal{X}$ to the space of binary strings, while the decoder maps from binary strings back into a reconstruction alphabet $\hat{\mathcal{X}}$.

The rate of a source code is the average length of the binary strings generated by the encoder. Because in general it is impossible to have perfect translation at a finite rate, the performance of a source code is evaluated relative to some per-letter distortion function ρ by looking at

$$\sup_{t>0} E[\rho(X_t, \hat{X}_t)]$$

or

$$\lim_{N\to\infty} E[\frac{1}{N}\rho(X_1^N, \hat{X}_1^N)]$$

This is referred to as the distortion of the code. The distortion measure ρ is usually chosen for tractability and as a way to approximate the sense of fidelity we are interested in for the real world situation. Popular distortion measures that we will be interested in are squared error $\rho(X, \hat{X}) = |X - \hat{X}|^2$ and general moments $\rho_\eta(X, \hat{X}) = |X - \hat{X}|^\eta$ where $\eta > 0$.

2.1.3 Noisy Channels

Noisy channels are used to approximate real world communication media that are unreliable or introduce noise. We focus on memoryless channels in this thesis.

Definition 2.1.2 *A* memoryless noisy channel *is a stochastic kernel $P(a|b)$ from an input $a \in \mathcal{A}$ to an output $b \in \mathcal{B}$*

Erasure Channel

A popular discrete-time channel is the erasure channel. It is often used to model channels in which the receiver can detect when an error has occurred.

Definition 2.1.3 *The* binary erasure channel *with erasure probability e is a memoryless noisy channel with $\mathcal{A} = \{0, 1\}$, $\mathcal{B} = \{0, 1, \emptyset\}$ and $P(a = b) = (1 - e)$ and $P(b = \emptyset) = e$.*

AWGN Channel

Another important channel is the Additive White Gaussian Noise (AWGN) channel.

Definition 2.1.4 *A* scalar AWGN channel with variance K_V and power constraint P *is a memoryless noisy channel with $\mathcal{A} = \mathcal{B} = \Re$ and $P(B = a + V|a)$ distributed like a zero mean Gaussian random variable V with variance K_V. There is an additional constraint on the input: $E[A^2] \leq P$*

The power constraint is needed to prevent degenerate solutions. Without it, solutions might end up using arbitrarily large signals that would completely dominate the effect of the additive noise.

2.1.4 Channel Codes

A channel code is an encoder/decoder pair $(\mathcal{E}, \mathcal{D})$ that can be wrapped around a noisy channel. The encoder maps strings of bits $\{S_t\}$ into channel inputs from the set $\mathcal{A}$, while the decoder maps channel outputs from the set $\mathcal{B}$ back into strings of bits $\{\hat{S}_t\}$. The correspondence between input bitstream $\{S_t\}$ and the output bitstream $\{\hat{S}_t\}$ is specified by the reconstruction profile r_1^∞.

The rate of the channel code is the number of bits the encoder takes in per unit time, which is the same as the number of bits the decoder outputs per unit time. While the goal is to reproduce the input bits perfectly at the output (generally with some delay determined by r_1^∞), this is not always possible. To measure the effectiveness, we need to define the probability of error.

Definition 2.1.5 *The* probability of error $P_{error}(\mathcal{E}, \mathcal{D}, r_1^\infty)$ *is the supremum over* $i > 0$ *of the probability of error in the* i*-th bit:*

$$P_{error}(\mathcal{E}, \mathcal{D}, r_1^\infty, i) = P(\hat{S}_i \neq S_i)$$

Block Codes

An important class of channel codes that are traditionally studied are block codes.

Definition 2.1.6 *For non-negative integers* R_{in} *and* R_{out}*, a* (R_{in}, R_{out}) block channel encoder *is a function* $\mathcal{E}$ *from* $\{0,1\}^{R_{in}}$ *into* $\mathcal{A}^{R_{out}}$*. Similarly, a* (R_{in}, R_{out}) block channel decoder *is a function* $\mathcal{D}$ *from* $\mathcal{B}^{R_{out}}$ *into* $\{0,1\}^{R_{in}}$*. The block code has rate* $\frac{R_{in}}{R_{out}}$ *bits per channel use. The range of the encoder is called the set of codewords for the code.*

The maximum end-to-end delay of a block code is approximately the sum of the amount of time it takes to get R_{in} bits plus the amount of time it takes to send out R_{out} channel symbols. This is because the entire block of input bits needs to be ready before we can compute the codeword to be transmitted. Once the codeword has been computed, we must wait another block of time for it to be transmitted to the receiver who can then decode it instantly (ignoring any computational limitations).

2.2 Reliable Communication of Bits

One of the basic motivations behind information theory is to communicate bits reliably through noisy or constrained channels.

2.2.1 "Streaming" and Delay

Before we review existing senses of reliable communication, we need to discuss the role of delay to interpret situations in a "streaming" perspective. In our discrete-time streaming view, time starts at 0 and then goes on forever. Of course, time does not really go on forever for any foreseeable system that we encounter. Even our solar system is expected to come to an end at some point! Still, the infinite horizon view is an important modeling tool that enables us to deal with situations where the time horizon for the ongoing process is far longer than any particular duration or delay that we are interested in. Yet introducing a potential infinity within our model is not without some peril. This infinite domain for time can allow for paradoxical interpretations if we do not require that our codes have some sense of bounded end-to-end delay.

The Hotel Infinity

In [26], Gardner introduces a hypothetical "Hotel Infinity" with rooms labeled by the positive integers. He illustrates the seemingly paradoxical properties of infinite sets by showing how when the hotel has no vacancies, it is still possible to make room for any finite number of incoming guests by just shifting people to larger room numbers. The more fascinating fact is that it is possible to accommodate even an infinite number of guests by asking existing guests to move to a room with double the room number and then accommodating the infinite number of new guests in the odd numbered rooms that become vacant.

The Code Infinity

We can use a similar argument[1] to construct a code for the binary erasure channel which "reliably" transmits at *any rate* regardless of the probability of erasure for the channel and without any feedback.[2] Suppose bits are coming in at a rate R per unit time. Consider an infinite sequence of buffers $\{\mathcal{B}_i\}$ all initially empty. At time l:

1. Place the R bits that just arrived into every buffer
2. Let $j = \lceil \frac{-1+\sqrt{1+8l}}{2} \rceil$ and then $k = \frac{(j+1)^2-j-1}{2} - l + 1$
3. Transmit the oldest bit still waiting in buffer $\mathcal{B}_k$ and remove that bit from $\mathcal{B}_k$. If no bit is waiting, just sent a 0 and ignore it on the receiver.

The receiver knows exactly which bits are still awaiting transmission in each buffer and so can interpret every symbol that it receives. With this strategy, every buffer is visited an infinite number of times since this is just the explicit formula for visiting the buffers in the order $[1, 2, 1, 3, 2, 1, 4, 3, 2, 1, 5, 4, \ldots]$ forever. Each positive integer appears in an infinite number of spots in the list.

Since every buffer is visited an infinite number of times, every incoming bit is sent exactly once across the channel for each buffer. This means that every bit is sent an infinite number of times since there are an infinite number of buffers. Thus the receiver receives an infinite number of samples of every source bit and so with probability 1 can eventually decode every bit correctly![3]

Is it reasonable to interpret this code as achieving reliable transmission no matter how high the erasure probability? That seems like a silly interpretation since at any finite time, only a small fraction of the bits have even had a single opportunity for transmission. In fact, the minimum delay in this scheme increases with the bit position being considered.

We consider the delay of a code to be the supremum over the end-to-end delays for each bit position. It should be clear that to avoid situations like "The Code Infinity" we must require that the delay be bounded in some sense for every bit position in the infinite stream. While we will wish to look at limits as this end-to-end delay goes to infinity, actually infinite delay can make discussions of rate meaningless.

[1] Actually, our argument corresponds to a situation where we accomodate an infinite number of infinite families at the Hotel Infinity giving every family member her own room.

[2] This construction is related to Bertrand Russel's Paradox of Tristram Shandy as discussed in [10]. Tristram Shandy was a hypothetical immortal historian who would take one year to transcribe the events of a single day. Although he would fall further and further behind as time went on, paradoxically he would still eventually write a history for all time!

[3] Notice that this argument is not tied to the erasure channel and could work for any channel which is suitably ergodic.

2.2.2 Classical Notions Of Capacity

Noisy channels are traditionally characterized by different measures of capacity: "upto what rate can we transmit data reliably through the channel." The difference between notions of capacity is in terms of what is meant by "reliably." There are two main classical notions of this. Although these are traditionally discussed in terms of block-codes, we will implicitly use Lemma A.5.1 and give the definitions in our terms.

Definition 2.2.1 *Let $\mathcal{R}$ be a* reliable transmission property *which is a Boolean function and either true or false for a given system consisting of a channel encoder $\mathcal{E}$, channel decoder $\mathcal{D}$ (along with a reconstruction profile r_1^∞), and a noisy channel. If it is true, we say that $(\mathcal{E}, \mathcal{D})$ achieve $\mathcal{R}$ for that channel.*

The $\mathcal{R}$-capacity *$C_{\mathcal{R}}$ of a channel is the supremal rate R at which there exist $(\mathcal{E}, \mathcal{D})$ achieving $\mathcal{R}$ for that channel.*

$$C_{\mathcal{R}} = \sup\{R \,|\, \exists(\mathcal{E}, \mathcal{D}, r_1^\infty)\ Rate(\mathcal{E}, \mathcal{D}) = R, \mathcal{R}(\mathcal{E}, \mathcal{D}, r_1^\infty) = 1\}$$

This is an operational definition and is given in terms of the existence of encoders and decoders satisfying the desired property. It tells us what capacity means in terms of getting bits across a channel, not how to calculate its value for any specific channel.

Zero Error Capacity

The first notion of reliable transmission is a very strong one. In words, it says that after some delay T, we know the exact value for the transmitted bit without error. Mathematically:

$$\mathcal{R}_0(\mathcal{E}, \mathcal{D}, r_1^\infty) = \begin{cases} 1 & \text{if } \exists T, \text{Delay}(R, r_1^\infty) \leq T, P_{\text{error}}(\mathcal{E}, \mathcal{D}, r_1^\infty) = 0 \\ 0 & \text{otherwise} \end{cases}$$

The resulting $\mathcal{R}_0$-capacity corresponds to:

Definition 2.2.2 *[61] The* Shannon zero-error capacity *C_0 of a channel is the least upper bound of the rates at which the channel can be used to transmit data with a zero probability of error.*

$$C_0 = \sup\{R \,|\, \exists(T, \mathcal{E}, \mathcal{D}, r_1^\infty)\ P_{error}(\mathcal{E}, \mathcal{D}, r_1^\infty) = 0, Rate(\mathcal{E}, \mathcal{D}) = R, Delay(R, r_1^\infty) \leq T\}$$

Calculating zero-error capacity for even simple channels is in general an open problem usually studied in combinatorics and graph theory. [38]

Shannon Capacity

A weaker notion of reliable transmission allows for some probability of error:

$$\mathcal{R}_\epsilon(\mathcal{E}, \mathcal{D}, r_1^\infty) = \begin{cases} 1 & \text{if } \exists T, \text{Delay}(R, r_1^\infty) \leq T, P_{\text{error}}(\mathcal{E}, \mathcal{D}, r_1^\infty) \leq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

For every $\epsilon > 0$, we have a new notion of $\mathcal{R}_\epsilon$-capacity C_ϵ. If we want the probability of error to be arbitrarily small, it is natural to look at:

$$C = \lim_{\epsilon \to 0} C_\epsilon$$

Shannon's genius was to realize that this need not be the same as the zero error capacity. In fact, as the examples in Section 2.2.3 show, it can be much larger.

Definition 2.2.3 *[23] The* Shannon classical capacity C *of a channel is the least upper bound of the rates at which the channel can be used to transmit data with an arbitrarily small probability of error.*

$$C = \sup\{R \,|\forall \epsilon > 0\, \exists (T, \mathcal{E}, \mathcal{D}, r_1^\infty)\; P_{error}(\mathcal{E}, \mathcal{D}, r_1^\infty) \leq \epsilon, Rate(\mathcal{E}, \mathcal{D}) = R, Delay(R, r_1^\infty) \leq T\}$$

The definition we have given is an operational one. In this definition, just like the one for zero-error capacity, the channel encoder/decoder pair $\mathcal{E}, \mathcal{D}$ is used with a reconstruction profile r_1^∞ that has end-to-end delay less than or equal to T. The delay is shown as a function of only the reconstruction profile and the input rate since the channel's sampling time and offsets are considered to be given. The purpose of the delay is to have time to exploit laws-of-large-numbers. By inspection, we know $C_0 \leq C$.

One of the wonderful things about the Shannon classical capacity C is that it is possible to calculate it as the solution to a particular optimization problem depending only on the properties of the channel. For a memoryless channel, this is the famous maximum mutual information characterization. [23]

$$C = \sup_{P(A)} I(A; B) \tag{2.1}$$

where the random variables $A \in \mathcal{A}$ and $B \in \mathcal{B}$ are linked by the transition probability defining the channel. There has been a lot of work in getting similar characterizations to help us calculate capacity in general cases. [67]

The fact that Shannon classical capacity is computable by (2.1) and other explicit formulae also lets us prove many results about it. For example, it is well-known that the Shannon classical capacity of a memoryless channel is the same regardless of whether the codes are allowed access to feedback.[23]

2.2.3 Example Capacities

Equation (2.1) lets us calculate the capacities for various channels in closed form.

Binary Erasure

For the binary erasure channel with inter-sample time τ, it can be shown that the Shannon classical capacity of this channel is $\frac{1-e}{\tau}$ bits per unit time (or $1 - e$ bits per channel use) regardless of whether the encoder has feedback or not.[23] Furthermore, because a long string of erasures is always possible, the Shannon zero-error capacity of this channel is 0 as long as $e > 0$.

If noiseless feedback is available, the erasure channel has a particularly intuitive encoder which achieves capacity. The encoder maintains a first-in-first-out (FIFO) buffer and retransmits any bit that does not get through. As long as the input rate is sufficiently low, the input buffer will stay finite and every bit will make it through eventually. Formally,

$$\mathcal{E}_i(s_1^{\lfloor \frac{iR}{\tau} \rfloor}, b_1^{i-1}) = s_{\lfloor \frac{iR}{\tau} \rfloor - j_i(b_1^{i-1})} \tag{2.2}$$

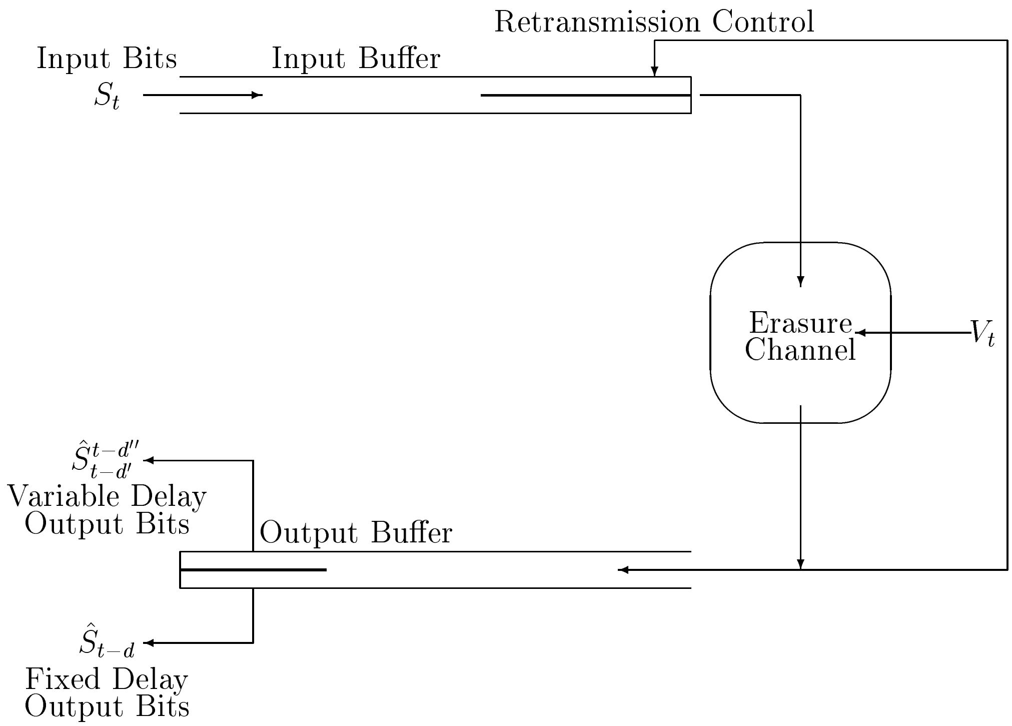

Figure 2-1: Coding for the erasure channel with noiseless feedback

where j is the current buffer size. We can define j recursively as follows: If $\lfloor \frac{iR}{\tau} \rfloor = 0$, then $j_i = 0$. Beyond that:

$$j_i(b_1^{i-1}) = \max(0, j_{i-1}(b_1^{i-2}) + \lfloor \frac{iR}{\tau} \rfloor - \lfloor \frac{(i-1)R}{\tau} \rfloor - 1 + \delta(b_{i-1}, \emptyset)) \tag{2.3}$$

where δ is the usual Kronecker delta function.

So the buffer shrinks by 1 if something useful was received last time and it grows according to the rate at which bits come in. Furthermore, since the buffer size depends only on the time and the received symbols b, the decoder can track it as well. Therefore, the decoder knows which transmission corresponds to which input bit and can give meaningful outputs. As long as the buffer size does not go to infinity, every bit is eventually successfully transmitted and received, even though we cannot predict in advance how much delay will be encountered by any given bit.

Additive White Gaussian Noise Channel

For the scalar AWGN channel with sampling time τ, variance K_V, and power constraint P, it can be shown that the Shannon classical capacity is $\frac{1}{2\tau}\log_2(1+\frac{P}{K_V})$ bits per unit time regardless of whether the encoder has feedback or not.[23] Traditionally, zero-error capacity has not been evaluated for continuous channels. ([4] is apparently the only exception!) But the same logic applies as before and a string of rare events is always possible, so the Shannon zero-error capacity of this channel is 0 as long as $K_V > 0$.

2.2.4 Trading off Delay and Errors

Our discussion in Section 2.2.1 already tells us that a sense of bounded delay is important for the notion of reliable transmission to be meaningful in the streaming context. In both the zero-error and Shannon capacities, this is satisfied since the codes are required to have a hard bound on delay for every bit position. While in zero-error capacity, the probability of error is required to be exactly 0, the sense of reliable transmission for Shannon capacity allows us to talk about a tradeoff between the required delay and the probability of error.

Error Exponents

Since Shannon capacity C is considered the limit of C_ϵ as $\epsilon \to 0$, it is natural to wonder how $C_\epsilon \to C$. Traditionally, we are interested in more than the tradeoff between rate R and probability of error ϵ. It is clear that to achieve smaller probability of error, we have to tolerate longer delays if we hold the rate constant. This happens whenever the zero error capacity does not equal the Shannon classical capacity. We could just as well define another function $d(\epsilon, R)$ of the rate R and probability of error ϵ that expresses the minimum delay required to achieve that particular (ϵ, R) performance pair.

The traditional tool to characterize this tradeoff between delay and probability of error is the reliability function which gives an "error-exponent" for a given rate. The reliability function is generally defined in terms of block codes.[23]

Definition 2.2.4 *A* reliability function $E(R)$ *for a particular channel is a function which relates the probability of error to the block length as follows:* $\exists \mathcal{E}_0, \mathcal{D}_0, R_{in}, R_{out}$ *such that* $P_{error} \leq 2^{-E(\frac{R_{in}}{R_{out}})R_{out}}$.

Traditionally, this has been considered a measure of code complexity rather than of delay *per se*. Block codes with larger block lengths were considered more complex than ones with short block-lengths. As a result, similar definitions have also been given in terms of the constraint length of convolutional codes.

While reliability functions are certainly very useful in characterizing the tradeoff between delay and probability of error, their tight connection to measures of encoder complexity like block length (R_{out}) is problematic if what we are really interested in is end-to-end delay. Recall that the end-to-end delay for a block channel code is at least $\tau(2R_{out} - 1 - \frac{R_{out}}{R_{in}})$ units of time. This means that in terms of delay, the exponent should be considered half of $E(R)$. Also, the fact that both R_{in} and R_{out} must be integers means that delay cannot even be defined for irrational rates and is in fact very discontinuous as a function of rate.

For example, assume $\tau = 1$. $R = \frac{1}{2} = 0.5$ and $R = \frac{233}{467} = 0.498929\ldots$ are very close to each other as real numbers, but since 467 is prime, any block code for the second rate must have a block length that is an integral multiple of 467 and a delay of at least 932 units! For evaluating performance at moderate values of delay, it matters little that the two error exponents might differ very slightly.

The error exponents given in terms of the constraint length or number of states of a convolutional code are even less illuminating when it comes to evaluating the tradeoff between delay and probability of error. This comes from the fact that a short constraint length or a small number of states can still introduce long-range dependencies in the codewords. As a result, optimum Viterbi decoding of convolutional codes can introduce long delays, even with infinite speed computations. The same is true for sequential decoding. However, Forney has studied the tradeoff of probability of error with respect to delay if we force the decoder to make a decision after a certain delay d. By introducing a way of getting block codes from convolutional ones, he was able to show that the block coding error exponents govern the tradeoff with delay, even though the error exponents relative to the constraint length can be much bigger. [19, 20] Bounds of a similar spirit extending to nonblock codes are also available in [50].

Other Ideas

The main other work relating to trading off delays and errors is that of Krich [42]. This, along with his dissertation [40], discusses weighting the probability of error on every bit by a function $f(d)$ which depends on the delay experienced by that bit. Although he is motivated by the stochastic control problem as we are, he restricts attention throughout his work to block codes. He shows that unlike the unweighted average probability of error case where the best codes have very long block lengths, with many weighting functions the optimal block-lengths are finite to balance the cost of increasing delay for every bit with the benefits of higher accuracy.

Although Krich introduces a weighting function, he never uses it to come up with a new sense of reliable transmission. Instead, he uses it to help guide the choice of block-codes designed to achieve Shannon's sense of reliable transmission. As section 2.4.2 will demonstrate, for unstable processes the Shannon sense of reliable transmission is not good enough and hence Krich's approach of using block-codes will not suffice for us.

2.3 Communicating Sources over Channels

Now, we will attempt to put together the traditional results on communication over noisy channels with the problem of tracking sources. First, we will recall the results for stable Markov sources with $\|A\| < 1$.

For convenience, we will assume that W_t is zero mean. Notice that by our definition, the stable source is tracked by even the trivial zero-rate process $\hat{X}_t = 0$ since X_t has asymptotic variance equal to $\frac{\sigma^2}{1-A^2} < \infty$. But if we are allowed more rate in encoding, we can usually do even better.

2.3.1 Source Coding

Definition 2.3.1 *For a random source $\{X_t\}$, define the operational rate-distortion function $R^{oper}(D)$ by:*

$$\inf\left\{R \,\middle|\, \forall\epsilon > 0, \exists(n, F_n^R, G_n^R), \lim_{i\to\infty} E\left[\frac{1}{n}\sum_{j=1}^{n}\rho(X_{in+j}, \left(G_n(F_n(X_{in+1}^{(i+1)n}))\right)_j)\right] < D + \epsilon\right\}$$

where (F_n^R, G_n^R) is a block source-encoder and source-decoder pair with Rate$(F_n^R, G_n^R) = R$ and block-length n.

Notice that the operational rate-distortion function above is defined in terms of block-codes. For memoryless processes, Shannon showed that the rate-distortion function can also be computed as the solution to a mutual information optimization problem.[62]

$$R(D) = \inf_{P(\hat{X}|X):E[\rho(X,\hat{X})]\leq D} I(X;\hat{X}) \tag{2.4}$$

For processes which are not i.i.d., we can consider them as the limit of larger and larger blocks and thereby get an asymptotic expression for the rate-distortion function as:

$$R(D) = \lim_{N\to\infty} \frac{1}{N} \inf_{P(\hat{X}_1^N|X_1^N):E[\rho^N(X^N,\hat{X}^N)]\leq ND} I(X_1^N;\hat{X}_1^N) \tag{2.5}$$

For stationary ergodic processes and a general class of distortion measures [6], $R^{\mathrm{oper}}(D) = R(D)$. The basic idea is that such processes have fading memories. Thus, disjoint long blocks have roughly the same distribution to each other and the dependencies between the long blocks are very slight. This flavor of result can be extended to some more ill-behaved processes as well [30], though the interpretation in such cases is a little subtle.

2.3.2 The Separation Theorem

The keystone result of Shannon's program is that the restriction to binary for source coding is justified since any rate/distortion performance that can be achieved with nonbinary codes can also be achieved with binary ones. This is called the separation theorem or information transmission theorem:

Theorem 2.3.1 *For a channel, independent stable source $\{X_t\}$, and per-letter distortion measure $\rho(X,\hat{X})$, the performance of any pair of joint source-channel encoder/decoder pairs $\mathcal{E},\mathcal{D}$ is bounded as follows: $E(\rho(X,\hat{X})) \geq R^{-1}(C)$ where $R^{-1}(C)$ is the distortion-rate function (the inverse of the rate-distortion function) evaluated at the Shannon capacity C of the channel. Moreover, we can get arbitrarily close to $R^{-1}(C)$ by using a good source coder followed by a good channel code.*

The traditional proofs rely on the fact that we have characterizations of both Shannon capacity C and rate-distortion function $R(D)$ in terms of mutual information optimization problems. The most sophisticated existing theorems in this regard are those found in [66], but their interpretation is subtle in nonstationary cases. They do not apply within a truly streaming view where the source is realized over time rather than all at once.

2.4 Trying Classical Ideas with Unstable Sources

The stable case in Theorem 2.3.1 seems to contrast sharply with Proposition 1.2.1 and Corollary 1.2.1 which assert the nonexistence of any block codes capable of tracking our simple random walk. However, the existence of the source encoder (1.1) and decoder (1.2) pair which does track our unstable source might give us some hope. Let us see what happens if we try to combine these with codes satisfying the traditional notions of channel capacity and reliable transmission.

2.4.1 Zero-Error Capacity

It is easy to see that if the zero-error capacity $C_0 > 1$, then we can find a delay T and $(\mathcal{E}_z,\mathcal{D}_z)$ such that we can transmit 1 bit without error every time step. Then, by using the source encoder F_t from equation (1.1) to generate the bits $\{S_t\}$, by time t we have $\hat{S}_1^{t-T} = S_1^{t-T}$ available at the output of the channel decoder. By combining this with the the source decoder G from equation (1.2) we can get $\hat{S}_{t-T} = X_{t-T}$. Even if we use this value as our estimate $\hat{X}_t$ for time t itself, we know by the nature of the source that $(X_t - X_{t-T})^2 \leq T^2$. So we can track the source over the channel.

2.4.2 Shannon Classical Capacity

The question is more interesting if $C_0 = 0$ as it is for many well-known channels like the binary symmetric channel, the binary erasure channel, or the power-constrained AWGN channel. Consider a case where $C > 1$. Then, we know by the definition of Shannon classical capacity that we have a delay T and $\mathcal{E}$, $\mathcal{D}$ such that we can transmit 1 bit every time step. If we once again use the source encoder F_t from (1.1) to generate the bits $\{S_t\}$, by time t we have $\hat{S}_1^{t-T}$. We can use the source decoder G from (1.2) to get $\hat{X}_{t-T} = \sum_{i=1}^{t-T}(2\hat{S}_i - 1)$. But we also have a probability of error ϵ and so it makes sense to look at the error: $(X_{t-T} - \hat{X}_{t-T}) = \sum_{i=1}^{t-T} 2(S_i - \hat{S}_i)$.

From the existing theorems about channel capacity, all we know about this process is that $S_i - \hat{S}_i = 0$ with probability at least $1 - \epsilon$. To see that this is not enough for $\{\hat{X}_t\}$ to track $\{X_t\}$, consider the possibility that $S_i - \hat{S}_i = \pm 1$ with probability $\frac{\epsilon}{2}$ each and that they are i.i.d. In that case, $E[\rho(X_t, \hat{X}_t)] = \sum_{i=1}^{t} 4E[(S_i - \hat{S}_i)^2] = 4\epsilon t$. Clearly, this has no finite bound and grows to infinity, no matter how small we make ϵ as long as it is not exactly zero.

The problem is that as time goes on, the number of errors made in the past can only grow. Unlike the stable case where the past gradually gets less and less important, for unstable processes the past continues to strongly influence the present. Since the past errors corrupt current estimates, this means that we slowly lose track of where we should be. Figure 2-2 shows what happens in the case where $A > 1$. As soon as a single uncorrected bit error occurs, the estimates diverge exponentially from the true process. The issue here is not that the capacity is too small, but that the very sense of reliable transmission itself is not adequate. Philosophically speaking, this means that the kinds of "bits" that we are using in our source encoders for the simple random walk are somehow incompatible with the Shannon classical capacity.

2.4.3 Binary Erasure Channel With Feedback

If we are using a binary erasure channel with the encoder allowed access to noiseless feedback, then we can use encoder (2.2) to get every bit through eventually. We know that the delay experienced by any given bit is the time it has to wait in the encoder's queue. If τ is the time between channel uses, as long as the incoming rate is less than $\frac{1-e}{\tau}$, the encoder's queue stays stable and its length is random and can be bounded by an exponential distribution. Since the squared-error distortion for our simple random walk is at most T^2 with a delay of T, we know that its expectation will stay finite for all time if the queue is stable. This is true even though the binary erasure channel with feedback has no strict zero error capacity! So zero-error capacity is not necessary, even though it is sufficient.

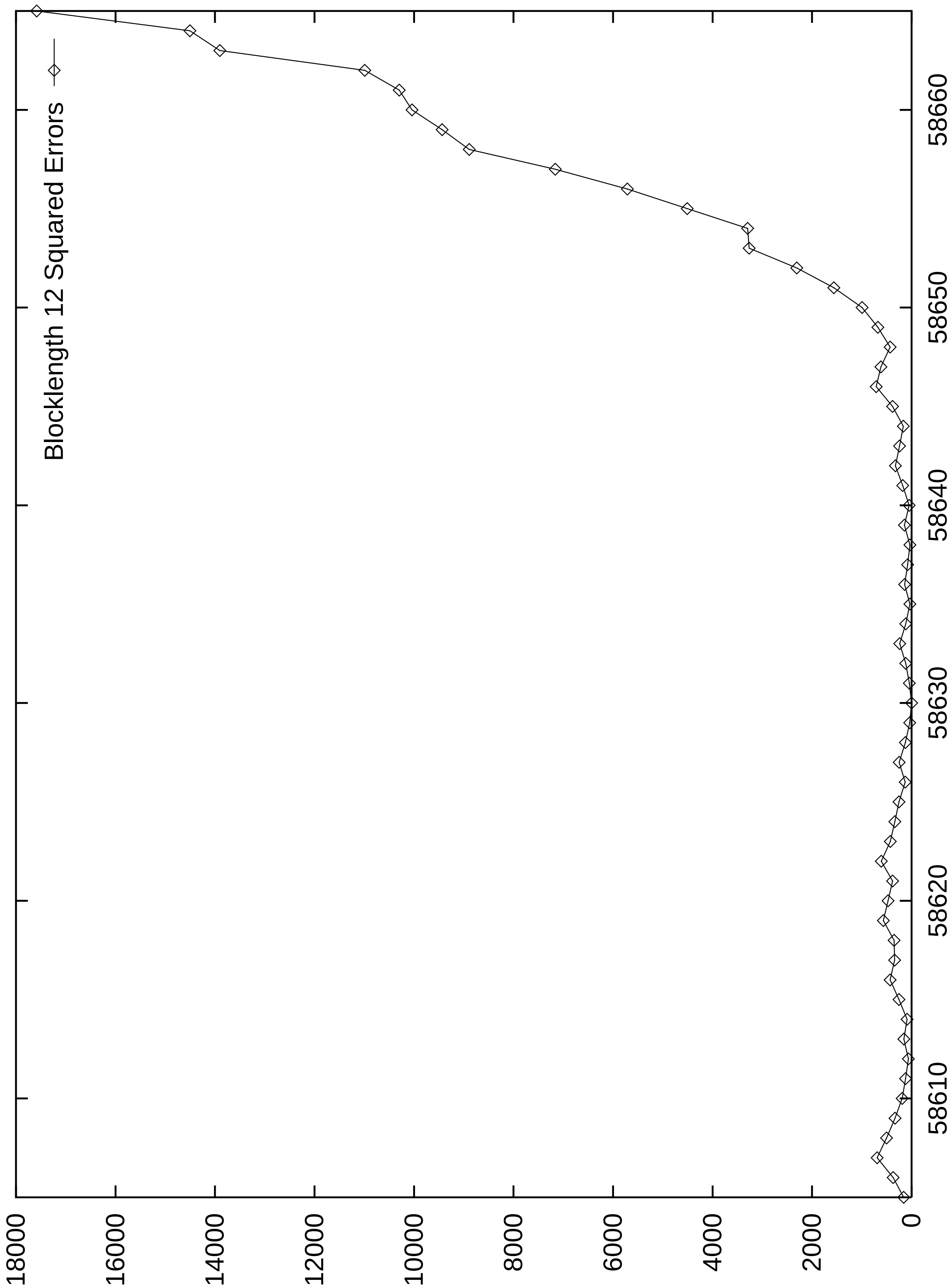

Figure 2-2: Simulation of distortion through time for a source code cascaded with a blocklength 12 classical channel code.

Chapter 3

Sensitivity Of Source Coding

In this chapter, we formalize the notion of sensitivity in source codes. By introducing a worst-case formulation for the accumulated effect of bit errors in distortion, we distinguish between strong and weak codes. After showing that all standard block codes are strong, we prove that any source code for an unstable Markov process must be weak.

3.1 The Meaning of "Bits"

We can be more precise with the "meaning of a bit" when the bit in question results from encoding a source. The root notion of meaning in such contexts is represented by the per-letter distortion measure ρ on the source and reconstruction alphabets. To understand what any particular bit "means," we need to pull ρ through the encoder/decoder pair down to the level of bits. We want to know the effect on the distortion of getting any particular bit wrong. Even though ρ itself is assumed to be per-letter and hence "local," the memory in the source encoders and decoders can result in the induced meaning of a bit varying with time.

In general, the reconstruction $\hat{x}_t$ can depend on the bits s_1^∞ in some complicated way. Because of this, we will abuse notation slightly and use $\hat{x}_t(s_1^\infty)$ to refer to the decoding function evaluated for the reconstruction of the source at time t on the stream s_1^∞. $\hat{x}_t$ will therefore be used to refer to both values for the random variable $\hat{X}_t$ and for the decoding rule itself. Consider the bit at position j. We can consider two different $\hat{x}_t$ based on whether s_j is zero or one. Let:

$$\Delta(j,t,s_1^\infty,x_t) = |\rho(x_t,\hat{x}_t(s_1^\infty)) - \rho(x_t,\hat{x}_t(s_1,\ldots,s_{j-1},(1-s_j),s_{j+1}^\infty))|$$

$\Delta(j,t,s_1^\infty,x_t)$ is implicitly a function of the distortion measure and source decoder. It measures the sensitivity of the reconstruction $\hat{x}_t$ to a particular bit given all the others. By taking an expectation over the source statistics, we can define:

$$\bar{\Delta}(j,t) = E_{X_1^\infty}\left[\Delta(j,t,S_1^\infty = F(X_1^\infty),X_t)\right]$$

$\bar{\Delta}(j,t)$ is therefore implicitly a function of the source distribution, distortion measure, source encoder, and source decoder. It measures the expected sensitivity of the reconstructions at time t to the value of bit j. However, it only deals with the propagated effect of a single bit error and does not consider the possible interactions of accumulated errors.

At this point, there are two possibilities on how to proceed. The initial thought might be to consider the effect of accumulated errors if only a small fraction of bits get changed. The hope would then be to design codes that could tolerate a small number of bits being changed. There are two possibilities for how we could formulate the problem of only a fraction of bits changing. We could go for a “worst-case” formulation in which an adversary could flip a fraction of bits at will and then take the maximum over the resulting distortions. Alternatively, we could take an “average-case” formulation in which a fraction of the bits are flipped at random and then look at the expected change in distortion. Both these formulations require us to introduce a new parameter for the fraction of bit flips. Furthermore, the average case formulation is equivalent to asking the question of whether the code could be successfully communicated over a binary symmetric channel with specified crossover probability. This suggests that looking at the problem of sensitivity to a fraction of changes mixes the issue of transmission over a noisy channel with the issue of sensitivity to errors in the past. So to focus on the issue of sensitivity to errors in the past, we concentrate on getting upper bounds to sensitivity.

3.2 Error Accumulation

To upper bound the effect of an accumulation of errors, we will define two new implicit functions of source codes:

$$\begin{aligned}
\Delta^+(j,t) &= E_{X_1^\infty}\left[\sup_{\tilde{S}_j^\infty}\rho(X_t,\hat{x}_t((F(X_1^\infty))_1^{j-1},\tilde{S}_j^\infty)) - \rho(X_t,\hat{x}_t(F(X_1^\infty)))\right] \\
\Delta^-(j,t) &= E_{X_1^\infty}\left[\sup_{\tilde{S}_1^j}\rho(X_t,\hat{x}_t(\tilde{S}_1^j,(F(X_1^\infty))_{j+1}^\infty)) - \rho(X_t,\hat{x}_t(F(X_1^\infty)))\right]
\end{aligned}$$

$\Delta^+(j,t)$ measures the sensitivity of X_t on all the bits from position j on while $\Delta^-(j,t)$ is for all the bits up to j. We can express this sensitivity purely in terms of delay for a rate R source code with:

$$\begin{aligned}
\Delta^+(d) &= \sup_{t>0}\Delta^+(\lfloor R(t-d)\rfloor,t) \\
\Delta^-(d) &= \sup_{t>0}\Delta^-(\lfloor R(t-d)\rfloor,t)
\end{aligned}$$

By inspection, any source code with a finite upper bound d_{max} on the delay of reconstruction has the property that $\Delta^+(-d)=0$ whenever $d>d_{max}$. This is because changing the bits far in the future has no effect on the reconstruction now.

Now, we can also bound these sensitivity measures for block source codes:

Lemma 3.2.1 *Let (F,G) be any rate $R=\frac{R_{in}}{R_{out}}$ block source code. Then $\Delta^+(j,t)=\Delta^-(k,t)=0$ for all $k<\left\lfloor\frac{t-1}{R_{in}}\right\rfloor R_{out}$ and $j>\left\lceil\frac{t-1}{R_{in}}\right\rceil R_{out}$. Thus $\lim_{d\to+\infty}\Delta^-(d)=0$ as well as the usual $\lim_{d\to-\infty}\Delta^+(d)=0$.*

If $\sup_{t>0}E_{X_t}\left[\max_{\tilde{S}_1^{R_{out}}\in\{0,1\}^{R_{out}}}\rho(X_t,\left(G(\tilde{S}_1^{R_{out}})\right)_{t \bmod R_{in}})\right]\leq K<\infty$ then furthermore $\Delta^+(j,t)\leq K$ as well as $\Delta^-(j,t)\leq K$. Thus, $\Delta^+(d)\leq K$ as well as $\Delta^-(d)\leq K$.

Proof: The first part is obvious since it is a block code and hence the decoded $\hat{X}_t$ only depends on the appropriate block of R_{out} bits. So:

$$
\begin{aligned}
& \Delta^+(j,t) \\
= \;& E_{X_1^\infty}\left[\sup_{\tilde{S}_j^\infty} \rho(X_t, \hat{x}_t((F(X_1^\infty))_1^{j-1}, \tilde{S}_j^\infty)) - \rho(X_t, \hat{x}_t(F(X_1^\infty)))\right] \\
= \;& E_{X_1^\infty}\left[\sup_{\tilde{S}_j^\infty} \rho(X_t, \hat{x}_t((F(X_1^\infty))_1^{j-1}, \tilde{S}_j^\infty)) - \rho(X_t, \left(G(F(X_{\left\lfloor \frac{t-1}{R_{in}} \right\rfloor R_{in}}^{(\left\lfloor \frac{t-1}{R_{in}} \right\rfloor + 1)R_{in}}))\right)_{t \bmod R_{in}})\right]
\end{aligned}
$$

But the block-code's limited dependencies mean that the first term in the expectation equals the second one regardless of the $\tilde{S}$ values and hence the sensitivity is zero. And analogously for $\Delta^-(k,t)$. The second part is also clear since:

$$
\begin{aligned}
\Delta^+(j,t) &= E_{X_1^\infty}\left[\sup_{\tilde{S}_j^\infty} \rho(X_t, \hat{x}_t((F(X_1^\infty))_1^{j-1}, \tilde{S}_j^\infty)) - \rho(X_t, \hat{x}_t(F(X_1^\infty)))\right] \\
&\le E_{X_1^\infty}\left[\sup_{\tilde{S}_1^\infty} \rho(X_t, \hat{x}_t(\tilde{S}_1^\infty)) - \rho(X_t, \hat{x}_t(F(X_1^\infty)))\right] \\
&\le E_{X_1^\infty}\left[\sup_{\tilde{S}_1^\infty} \rho(X_t, \hat{x}_t(\tilde{S}_1^\infty))\right] \\
&= E_{X_t}\left[\sup_{\tilde{S}_1^\infty} \rho(X_t, \hat{x}_t(\tilde{S}_1^\infty))\right] \\
&= E_{X_t}\left[\sup_{\tilde{S}_1^{R_{out}}} \rho(X_t, \left(G(\tilde{S}_1^{R_{out}})\right)_{t \bmod R_{in}})\right] \\
&\le K
\end{aligned}
$$

and similarly for $\Delta^-(j,t)$.

The results for $\Delta^-(d)$ and $\Delta^+(d)$ follow immediately. □

This suggests a more general definition covering the properties established by Lemma 3.2.1 for block source codes.

Definition 3.2.1 *If a rate R source code (F,G) with maximum delay d_{max} has $\Delta^-(d)$ finite for all d and $\lim_{d\to\infty} \Delta^-(d) = 0$, we say that the source code is* strong. *The resulting bits in the encoding S_1^∞ are also called* strong bits.

As we can see, Lemma 3.2.1 tells us that all block source codes are strong. With this definition, we can see that being strong is a sufficient condition for being able to cascade a particular source code with a standard Shannon channel encoder:

Theorem 3.2.1 *If a rate R source code (F,G) is* strong, *then $\forall \epsilon > 0$ and noisy channels with Shannon classical capacity $C > R$, there exists a delay d' such that the source code can be cascaded with an appropriate channel encoder/decoder to achieve expected per-letter distortion within ϵ with end-to-end delay bounded by d'.*

Proof: Given ϵ, we can pick a d_ϵ such that $\Delta^-(d_\epsilon) < \frac{\epsilon}{2}$ since the source code is strong. Now, let $\epsilon' = \frac{\epsilon}{2(d_\epsilon + d_{max})\Delta^-(-d_{max})}$ and choose a rate R channel code which has a per-bit probability of error of ϵ'. Set $d' = d_{max} + T$ where T is the delay introduced by the channel code. To see that cascading the source code with this channel code achieves the desired performance, let $\check{X}_t$ denote the final reconstruction of the cascaded system:

$$
\begin{aligned}
E\left[\rho(X_t, \check{X}_t)\right] &\leq E[\rho(X_t, \hat{X}_t)] + \Delta^-(d_\epsilon) + P(S_{\lfloor R(t-d_\epsilon)\rfloor}^{\lfloor R(t+d_{\max})\rfloor} \neq \hat{S}_{\lfloor R(t-d_\epsilon)\rfloor}^{\lfloor R(t+d_{\max})\rfloor})\Delta^-(-d_{max}) \\
&< E[\rho X_t, \hat{X}_t] + \frac{\epsilon}{2} + \epsilon'(d_{max} + d_\epsilon)\Delta^-(-d_{max}) \\
&< E[\rho X_t, \hat{X}_t] + \epsilon
\end{aligned}
$$

The end-to-end delay of the cascaded system is just the sum of the source code delay d_{max} and the channel code delay T. $\square$

Theorem 3.2.1 covers the case of strong source codes. From the definition of strong codes, we can see that this can cover most standard codes for fading-memory processes. However, to deal with other sorts of processes we will need:

Definition 3.2.2 *If a rate R source code (F, G) with maximum delay d_{max} has $\Delta^-(d)$ infinite for any d, we say that the source code is* weak. *The bits in the resulting encoding S_1^∞ are also called* weak bits.

We have a simple lemma which connects $\Delta^+(d)$ to our definition of weakness.

Lemma 3.2.2 *If a rate R source code (F, G) with maximum delay d_{max} has $\lim_{d\to\infty} \Delta^+(d) = \infty$, the source code is weak.*

Proof: Pick an arbitrarily large T. By the definition of limits, there exists a d_T such that $\Delta^+(d_T) \geq T$. Thus, there exists a t_T such that $\Delta^+(\lfloor Rt_T \rfloor - d_T, t_T) \geq T$. But then $\Delta^-(\lfloor Rt_T \rfloor + d_{max}, t_T) \geq T$ as well since it is a supremum over a strictly larger set. So $\Delta^-(d_{max}) \geq T$. But since T was arbitrarily high, this means that $\Delta^-(d_{max}) = \infty$ and the source code is weak. $\square$

Unlike strong bits, which Theorem 3.2.1 tells us are in some sense all alike, weak bits can be differentiated from each other by the rate at which $\Delta^+(d)$ goes to infinity as d increases. This is roughly analogous to the situation in computational complexity. The worst-case computational requirements for most interesting algorithms go to infinity as the problem size increases. But algorithms differ in the rate at which they do so.

As an example, we know from computational complexity theory that although the straightforward recursive algorithm for computing Fibonacci numbers takes exponential time as N gets larger, there is a simple iterative algorithm which works in linear time by building a table. Furthermore, if we assume the ability to do real number computations in unit time, it is possible to calculate the N-th Fibonacci number in constant time using the properties of the golden mean.

3.3 Scalar Markov Processes

For the stable case $|A| < 1$, we already know that strong codes exist. In the following two sections, we will illustrate the properties of weak codes by looking at the simple random walk and then the general case of exponentially unstable processes.

3.3.1 Simple Random Walk

For a contrast from the block-coding case as demonstrated by Lemma 3.2.1, let us take a look at the code from equation (1.1) and its corresponding decoder.

Proposition 3.3.1 *For the rate* 1 *code given by equations (1.1) and (1.2),* $\Delta^-(d) = \infty$ *for all* d *while* $(d+1)^2 \leq \Delta^+(d) \leq 4((d+1))^2$ *if* $d \geq 0$ *and zero otherwise.*

Proof: A straightforward calculation:

$$
\begin{aligned}
\Delta^+(d) &= \sup_{t>0} \Delta^+(t-d,t) \\
&= \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}_{t-d}^\infty} \rho(X_t, \hat{x}_t((F(X_1^\infty))_1^{t-d-1}, \tilde{S}_{t-d}^\infty)) - \rho(X_t, \hat{x}_t(F(X_1^\infty)))\right] \\
&= \sup_{t>0} E_{S_1^\infty}\left[\sup_{\tilde{S}_{t-d}^\infty}\left(\sum_{t-d}^{t}(2S_i - 1) - (2\tilde{S}_1 - 1)\right)^2 - 0\right] \\
&= \sup_{t>0} E_{S_1^\infty}\left[\sup_{\tilde{S}_{t-d}^\infty}\left(2\sum_{t-d}^{t}(S_i - \tilde{S}_1)\right)^2\right] \\
&\leq \sup_{t>0} 4(d+1)^2 \\
&= (2(d+1))^2
\end{aligned}
$$

and for the lower bound:

$$
\begin{aligned}
\Delta^+(d) &= \sup_{t>0} E_{S_1^\infty}\left[\sup_{\tilde{S}_{t-d}^\infty}\left(\sum_{t-d}^{t}(2S_i - 1) - (2\tilde{S}_1 - 1)\right)^2\right] \\
&= \sup_{t>0} E_{S_1^\infty}\left[\sup_{\tilde{S}_{t-d}^\infty}\left(|\sum_{t-d}^{t}(2S_i - 1)| + \sum_{t-d}^{t}(2\tilde{S}_1 - 1)\right)^2\right] \\
&\geq \sup_{t>0} \sup_{\tilde{S}_{t-d}^\infty}\left(\sum_{t-d}^{t}(2\tilde{S}_1 - 1)\right)^2 \\
&= (d+1)^2
\end{aligned}
$$

Since the source code is causal, the $\Delta^+(d)$ is zero for bit errors restricted to the future. The calculation for Δ^- is similar:

$$
\begin{aligned}
\Delta^-(d) &= \sup_{t>0} \Delta^-(t-d,t) \\
&= \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}_1^{t-d}} \rho(X_t, \hat{x}_t(\tilde{S}_1^{t-d}, (F(X_1^\infty))_{t-d+1}^\infty)) - \rho(X_t, \hat{x}_t(F(X_1^\infty)))\right] \\
&= \sup_{t>0} E_{X_1^\infty}\left[(2(t-d))^2 - 0\right] \\
&= \infty
\end{aligned}
$$

which shows the "weakness" that we have formalized above. □

Proposition 3.3.1 shows us that some very natural source codes are not strong. It is therefore natural to wonder whether the weakness of the source code (1.1) is just an artifact of their particular construction or whether it is something intrinsic to our source itself.

The following natural generalization of Proposition 1.2.1 answers the question:

Theorem 3.3.1 *All rate $R < \infty$ source codes with maximum delay $d_{max} < \infty$ that track the random walk process are weak regardless of the values of R and d_{max}. Furthermore, $\Delta^+(d)$ grows at least linearly with delay d under squared error distortion.*

Proof: Pick a delay d.

$$\begin{aligned}
\Delta^-(d) &= \sup_{t>0} \Delta^-(\lfloor R(t-d)\rfloor, t) \\
&= \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}_1^{\lfloor R(t-d)\rfloor}} \rho(X_t, \hat{x}_t(\tilde{S}_1^{\lfloor R(t-d)\rfloor}, (F(X_1^\infty))_{\lfloor R(t-d)\rfloor+1}^\infty)) - \rho(X_t, \hat{x}_t(F(X_1^\infty)))\right] \\
&= \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}_1^{\lfloor R(t-d)\rfloor}} \rho(X_t, \hat{x}_t(\tilde{S}_1^{\lfloor R(t-d)\rfloor}, (F(X_1^\infty))_{\lfloor R(t-d)\rfloor+1}^\infty))\right] \\
&\quad - \sup_{t>0} E_{X_1^\infty}\left[\rho(X_t, \hat{x}_t(F(X_1^\infty)))\right] \\
&\geq \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}_1^{\lfloor R(t-d)\rfloor}} \inf_{S_{\lfloor R(t-d)\rfloor+1}^\infty} \rho(X_t, \hat{x}_t(\tilde{S}_1^{\lfloor R(t-d)\rfloor}, S_{\lfloor R(t-d)\rfloor+1}^\infty))\right] - K \\
&= \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}_1^{\lfloor R(t-d)\rfloor}} \inf_{S_{\lfloor R(t-d)\rfloor+1}^{\lfloor R(t+d_{max})\rfloor}} \rho(X_t, \hat{x}_t(\tilde{S}_1^{\lfloor R(t-d)\rfloor}, S_{\lfloor R(t-d)\rfloor+1}^{\lfloor R(t+d_{max})\rfloor}))\right] - K \\
&\geq \sup_{t>0} \inf_{\hat{x}_t} E_{X_1^\infty}\left[\sup_{\tilde{S}_1^{\lfloor R(t-d)\rfloor}} \inf_{S_{\lfloor R(t-d)\rfloor+1}^{\lfloor R(t+d_{max})\rfloor}} \rho(X_t, \hat{x}_t(\tilde{S}_1^{\lfloor R(t-d)\rfloor}, S_{\lfloor R(t-d)\rfloor+1}^{\lfloor R(t+d_{max})\rfloor}))\right] - K \\
&\geq \sup_{t>0} \inf_{\hat{x}'_t} E_{X_1^\infty}\left[\inf_{S_{\lfloor R(t-d)\rfloor+1}^{\lfloor R(t+d_{max})\rfloor}} \left(X_t - \hat{x}'_t(S_{\lfloor R(t-d)\rfloor+1}^{\lfloor R(t+d_{max})\rfloor})\right)^2\right] - K \\
&= \sup_{t>0} \inf_{\hat{x}'_t} E_{X_1^\infty}\left[\inf_{S_{\lfloor R(t-d)\rfloor+1}^{\lfloor R(t+d_{max})\rfloor}} t\left(\frac{X_t}{\sqrt{t}} - \frac{\hat{x}'_t(S_{\lfloor R(t-d)\rfloor+1}^{\lfloor R(t+d_{max})\rfloor})}{\sqrt{t}}\right)^2\right] - K
\end{aligned}$$

The key is to notice that $\frac{X_t}{\sqrt{t}}$ tends to a zero-mean unit variance Gaussian as t gets large. Therefore, we can view $\frac{\hat{x}'_t}{\sqrt{t}}$ as a $R(d + d_{max}) + 1$ bit optimal quantizer for that Gaussian. By standard rate-distortion theory [6] we know that the best mean-squared error is $2^{-2(1+R(d+d_{max}))}$ so we can write:

$$\begin{aligned}
\Delta^-(d) &\geq \sup_{t>0} \inf_{\hat{x}'_t} E_{X_1^\infty}\left[\inf_{S_{\lfloor R(t-d)\rfloor+1}^{\lfloor R(t+d_{max})\rfloor}} t\left(\frac{X_t}{\sqrt{t}} - \frac{\hat{x}'_t(S_{\lfloor R(t-d)\rfloor+1}^{\lfloor R(t+d_{max})\rfloor})}{\sqrt{t}}\right)^2\right] - K \\
&\geq \sup_{t>0} t2^{-2(1+R(d+d_{max}))} - K \\
&= \infty
\end{aligned}$$

To see that the $\Delta^+(d)$ must grow at least linearly, we can go through a similar argument:

$$\begin{aligned}
\Delta^+(d) &= \sup_{t>0} \Delta^+(\lfloor R(t-d) \rfloor, t) \\
&= \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}^\infty_{\lfloor R(t-d)\rfloor}} \rho(X_t, \hat{x}_t((F(X_1^\infty))_1^{\lfloor R(t-d)\rfloor -1}, \tilde{S}^\infty_{\lfloor R(t-d)\rfloor})) - \rho(X_t, \hat{x}_t(F(X_1^\infty)))\right] \\
&= \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}^\infty_{\lfloor R(t-d)\rfloor}} \rho(X_t, \hat{x}_t((F(X_1^\infty))_1^{\lfloor R(t-d)\rfloor -1}, \tilde{S}^\infty_{\lfloor R(t-d)\rfloor}))\right] \\
&\quad - \sup_{t>0} E_{X_1^\infty}\left[\rho(X_t, \hat{x}_t(F(X_1^\infty)))\right] \\
&\geq \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}^\infty_{\lfloor R(t-d)\rfloor}} (X_t - \hat{x}_t((F(X_1^\infty))_1^{\lfloor R(t-d)\rfloor -1}, \tilde{S}^\infty_{\lfloor R(t-d)\rfloor}))^2\right] - K \\
&\geq \sup_{t>0} E_{X_1^\infty}\left[(X_t - X_{t-\lfloor d\rfloor})^2\right] - K \\
&= \lfloor d \rfloor - K \\
&\geq d - 1 - K
\end{aligned}$$

In the above, the fifth line is because the best mean-squared estimate for X_t is its conditional expectation given what we know. But X_t, as a random walk, is a martingale and so the best we can possibly do is use the true value for $X_{t-\lfloor d\rfloor}$. □

Theorem 3.3.1 tells us that the weakness of (1.1) is not an artifact and is fundamental to the source itself.

3.3.2 General $A > 1$

In this section, we extend Theorem 3.3.1 to establish a similar result for the exponentially unstable sources.

Theorem 3.3.2 *If the W_t are independent of each other for our scalar Markov source with parameter A, all finite expected distortion source codes have $\Delta^+(d)$ growing exponentially at least as fast as $A^{2d} = 2^{(2\log_2 A)d}$ when viewed under the squared-error distortion measure and more generally, as $A^{\eta d}$ under the η-distortion measure $\rho_\eta(X, \hat{X}) = |X - \hat{X}|^\eta$ where $\eta > 0$.*

Proof: Without loss of generality assume that W_t is zero mean. Now regardless of the rate R, we can pick a delay d and repeat the basic argument from the proof of Theorem 3.3.1:

$$\begin{aligned}
&\Delta^+(d) \\
&= \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}^\infty_{\lfloor R(t-d)\rfloor}} \rho(X_t, \hat{x}_t((F(X_1^\infty))_1^{\lfloor R(t-d)\rfloor -1}, \tilde{S}^\infty_{\lfloor R(t-d)\rfloor})) - \rho(X_t, \hat{x}_t(F(X_1^\infty)))\right] \\
&= \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}^\infty_{\lfloor R(t-d)\rfloor}} \rho(X_t, \hat{x}_t((F(X_1^\infty))_1^{\lfloor R(t-d)\rfloor -1}, \tilde{S}^\infty_{\lfloor R(t-d)\rfloor}))\right] \\
&\quad - \sup_{t>0} E_{X_1^\infty}\left[\rho(X_t, \hat{x}_t(F(X_1^\infty)))\right]
\end{aligned}$$

$$
\begin{aligned}
&\geq \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}^\infty_{\lfloor R(t-d)\rfloor}} |X_t - \hat{x}_t((F(X_1^\infty))_1^{\lfloor R(t-d)\rfloor -1}, \tilde{S}^\infty_{\lfloor R(t-d)\rfloor})|^\eta\right] - K \\
&\geq \sup_{t>0} E_{X_1^\infty}\left[|X_t - A^{\lfloor d\rfloor}X_{t-\lfloor d\rfloor}|^\eta\right] - K \\
&= \sup_{t>0} E_{X_1^\infty}\left[|\sum_{i=0}^{\lfloor d\rfloor} A^i W_{t-i}|^\eta\right] - K \\
&\geq \sup_{t>0} E_{X_1^\infty}\left[|A^{\lfloor d\rfloor}W_{t-\lfloor d\rfloor}|^\eta\right] - K \\
&> MA^{\eta\lfloor d\rfloor} - K
\end{aligned}
$$

The steps are exactly the same as in the proof of Theorem 3.3.1, except for the last three steps. The best estimate for X_t given only information from d time steps ago is no better than the best prediction $A^d X_{t-d}$ if d is a positive integer, and in particular, we can never do better in prediction than the expected effect of the first driving noise term that we have no observations of. This gives rise to the positive M as above. □

In the next chapter, we will give a causal code which achieves the above bound on sensitivity. The immediate consequence of both Theorem 3.3.1 and 3.3.2 is that the same inherent weakness exists if we look at a downsampled version of the original source. If we are interested only in every l-th sample, the resulting process is defined by:

$$
\begin{aligned}
X_{lt} &= A^l X_{l(t-1)} + \sum_{j=0}^{l-1} A^{l-1-j} W_{t(l-1)+j} \\
&= A^l X_{l(t-1)} + \tilde{W}_t
\end{aligned}
$$

From Theorem 3.3.2, we know that the sensitivity grows as $A^{\eta d}$ if we assume the source evolves with one sample per unit time. By scaling time down by a factor of l while updating A to A^l, we get a sensitivity that grows as $(A^l)^{\eta \frac{d}{l}} = A^{\eta d}$ as well. The same argument also shows that the minimum rate per unit time required for tracking does not change if we downsample the source.

3.4 Discussion

In this section, we have established that there is a fundamental difference between strong and weak source codes. The strong codes eventually discount the past when trying to reconstruct the present while the effect of the past persists for the weak codes. Bitstreams emerging from weak codes can vary in the rate that the effect of errors can compound, and more importantly, for many sources and distortion measures there is a certain irreducible weakness that all codes must exhibit. The unstable scalar Markov sources exhibit just such a fundamental weakness.

The initial investigations in this chapter are just the beginning. Many questions remain to be answered. For encodings of vector sources, it seems clear that there might be substreams which have different inherent sensitivities and many encodings might be able to be split between an inherently weak core and a strong supplementary portion. A detailed study of the finer structure of the sensitivities within the encodings of complex sources should be undertaken. In addition, the ideas of sensitivity need to be extended into the context of lossless coding where there is no explicit distortion measure. I suspect that there is a purely "entropy based" formulation which will also capture the inherent sensitivity of certain random sources.

Furthermore, there is no reason to think that fundamental sensitivity is only interesting for signals modeled by linear Markov sources. We suspect that many real-world multimedia signals will also exhibit such phenomena, possibly in an approximate form when moderate delays are considered. It might also be a fundamental property of most variable rate or universal codes. In variable rate codes which generate self punctuating bitstreams, an early error might change the parsing of the whole stream, thereby having a lasting effect. In a universal code early errors might change the implicitly or explicitly estimated parameters for the source being encoded thereby altering the interpretations of all future bits. This is already being seen in practice [21] and needs to be better understood theoretically.

Chapter 4

Source Codes for Unstable Processes

As the previous chapter reviewed, source codes are a pair of mappings from the source alphabet to binary strings and then back again to a reconstruction alphabet. A source code is therefore a way of expressing "information" to some fidelity in the common language of binary strings. In this chapter, we will focus on the source from Definition 2.1.1, an unstable scalar Markov source $\{X_t\}$ driven by external noise $\{W_t\}$. Recall that this is:

$$\begin{aligned} X_0 &= 0 \\ X_t &= AX_{t-1} + W_t \end{aligned}$$

where the scalar $A \geq 1$ to make things nonstationary. The reconstruction alphabet is also the reals and the distortion measure ρ we use throughout is either the standard squared error distortion $\rho(X, \hat{X}) = |X - \hat{X}|^2$ or the more general η-distortion measure $\rho_\eta(X, \hat{X}) = |X - \hat{X}|^\eta$ with $\eta > 0$.

Before we can give theorems regarding information transmission over noisy channels in later chapters, we first need some understanding of how to encode unstable Markov sources into bits in the first place.

4.1 Causal Source Codes

The first encoders are in the spirit of (1.1) and we will assume that the W_t have bounded support: $-\frac{\Omega}{2} \leq W_t \leq \frac{\Omega}{2}$. It turns out that we do not require any other requirement on the driving noise process $\{W_t\}$ such as independence, ergodicity, or even stationarity.

Theorem 4.1.1 *For all $\epsilon > 0$, every scalar discrete-time unstable linear Markov process with parameter $a > 1$ driven by bounded noise $-\frac{\Omega}{2} \leq W_t \leq \frac{\Omega}{2}$, can be tracked by an encoder F^R and decoder G^R with rate $R \leq \log_2 A + \epsilon$. Moreover, there exists a constant ν depending only on (A, Ω, R) such that $\|X_t - \hat{X}_t\| \leq \nu$ for all t.*

Proof: We follow the spirit of [63] and encode the predictive error signal $X_{t+1} - A\hat{X}_t$.

Formally, we pick a $\nu > \frac{\Omega}{2}$. The idea is to keep the absolute value of the error always inside the of $[-\nu, \nu]$. Let $[-M_t, M_t]$ represent the best possible "box" in which the source decoder can currently bound the error signal. The predicted error signal $X_{t+1} - A\hat{X}_t$ is therefore known to be inside a new box $[-AM_t - \frac{\Omega}{2}, AM_t + \frac{\Omega}{2}]$ even without any additional

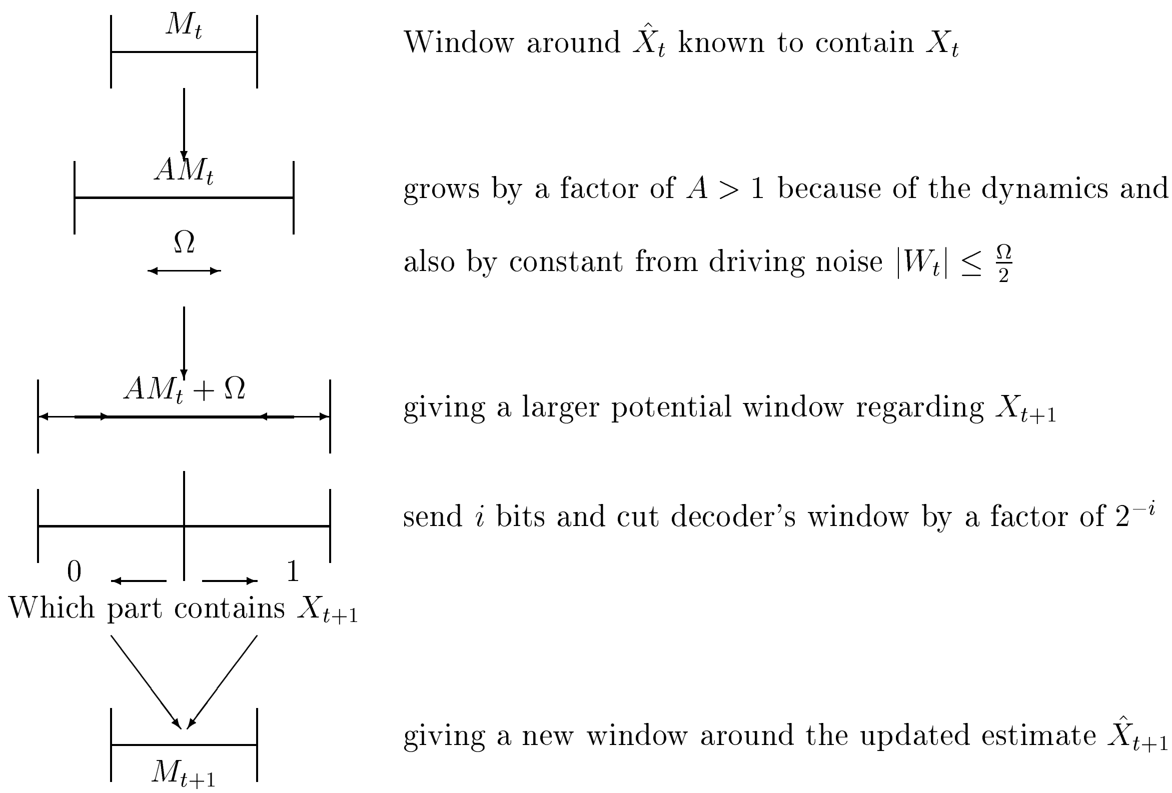

Figure 4-1: Causal predictive source code for unstable Markov process

information by simply predicting X_{t+1} by multiplying the existing $\hat{X}_t$ by A. If $AM_t + \frac{\Omega}{2} > \nu$, we need to send new bits of information to reduce the magnitude of the error. Since every bit we send can be used to cut the interval in half, we end up with the following recurrence relation:

$$M_{t+1} = \frac{AM_t + \frac{\Omega}{2}}{2^{i_{t+1}}}$$

where $i_{t+1} = \max\{0, \left\lceil \log_2 \frac{AM_t + \frac{\Omega}{2}}{\nu} \right\rceil\}$. The recurrence relation does not depend on the realization of the random process $\{X_t\}$ and so the M_t can be precomputed. They only depend on the initial value for M_0, the "box" size ν, the noise bound Ω, and the Markov parameter A. Even if the initial condition is exactly known, we will use the initial condition $M_0 = \nu$ for convenience.

The process is illustrated in Figure 4-1. i_{t+1} bits $S_1, S_2, \ldots, S_{i_{t+1}}$ are emitted at time $t+1$ and are chosen by the encoder so that

$$(X_{t+1} - A\hat{X}_t) \in [(AM_t + \frac{\Omega}{2})(-1 + 2^{-i_{t+1}} \sum_{j=0}^{i_{t+1}-1} S_j 2^j), (AM_t + \frac{\Omega}{2})(-1 + 2^{-i_{t+1}}(1 + \sum_{j=0}^{i_{t+1}-1} S_j 2^j))]$$

The estimates are generated as follows:

$$\hat{X}_{t+1} = A\hat{X}_t + (AM_t + \frac{\Omega}{2})(-1 + 2^{-i_{t+1}}(\frac{1}{2} + \sum_{j=0}^{i_{t+1}-1} S_j 2^j))$$

This encoder and decoder tracks the original random process since by construction $M_t < \nu$ for all t. All that remains is to calculate the rate at which bits emerge. By taking logs, it is easy to see that:

$$
\begin{aligned}
\log_2(M_{t+1}) &= \log_2 \frac{AM_t + \frac{\Omega}{2}}{2^{i_{t+1}}} \\
&= \log_2(AM_t + \frac{\Omega}{2}) - i_{t+1} \\
&= \log_2(M_t + \frac{\Omega}{2A}) + \log_2(A) - i_{t+1} \\
&= \log_2(M_t) + \log_2(1 + \frac{\Omega}{2AM_t}) + \log_2(A) - i_{t+1}
\end{aligned}
$$

By construction, we know that M_t is always between ν and $\frac{\nu}{2}$. Thus $\log_2 \nu \geq \log_2 M_t \geq \log_2 \nu - 1$ as well. Because this stays bounded, the rate at which bits flow out of the system $R = \lim_{t\to\infty} \frac{1}{t}\sum_{j=1}^{t} i_j$ must equal the rate that information flows in: $\lim_{t\to\infty} \frac{1}{t}\sum_{j=1}^{t} \log_2(A) + \log_2(1 + \frac{\Omega}{2AM_j})$. We can bound this as follows:

$$
\begin{aligned}
R &= \lim_{t\to\infty} \frac{1}{t}\sum_{j=1}^{t} \log_2(A) + \log_2(1 + \frac{\Omega}{2AM_j}) \\
&= \log_2(A) + \lim_{t\to\infty} \frac{1}{t}\sum_{j=1}^{t} \log_2(1 + \frac{\Omega}{2AM_j}) \\
&\leq \log_2(A) + \log_2(1 + \frac{\Omega}{A\nu})
\end{aligned}
$$

by using the lower bound $\frac{\nu}{2}$ for M_t. Similarly, by using the upper bound ν for M_t we get:

$$
\log_2(A) + \log_2(1 + \frac{\Omega}{A\nu}) \geq R \geq \log_2(A) + \log_2(1 + \frac{\Omega}{2A\nu})
$$

Since $\lim_{\nu\to\infty} \log_2(1 + \frac{\Omega}{A\nu}) = 0$, the rate can be made arbitrarily close to $\log_2(A)$. $\square$

It is easy to see that as long as $A \geq 1$, these source encoders are weak since they continuously build upon the prior estimates. The following theorem shows that the degree of weakness depends on A and achieves the fundamental sensitivity bound given in Theorem 3.3.2.

Theorem 4.1.2 *The source code given in Theorem 4.1.1 has $\Delta^+(d)$ growing exponentially as $A^{2d} = 2^{(2\log_2 A)d}$ when viewed under the squared-error distortion measure and more generally, as $A^{\eta d}$ under the η-distortion.*

Proof: This is relatively easy to see.

$$
\begin{aligned}
\Delta^+(d) &= \sup_{t>0} \Delta^+(\lfloor R(t-d)\rfloor, t) \\
&= \sup_{t>0} E_{X_1^\infty}\left[\sup_{\tilde{S}^\infty_{\lfloor R(t-d)\rfloor}} \rho(X_t, \hat{x}_t((F(X_1^\infty))_1^{\lfloor R(t-d)\rfloor - 1}, \tilde{S}^\infty_{\lfloor R(t-d)\rfloor})) - \rho(X_t, \hat{x}_t(F(X_1^\infty)))\right]
\end{aligned}
$$

The encoder always keeps the reconstruction within a ν box of the original signal as long as the bits are faithfully recovered. So we can think of starting at zero at time $t-d$ and trying to maximize the error in the next d time units. At the end, we will only be off by $\pm\nu$ at the most. But it is clear that maximizing the squared error between the reconstruction and the original starting at zero over d time units will give us (with appropriate constant M'):

$$\begin{aligned} M'(\sum_{i=0}^{d} A^i \nu)^\eta &= M'\nu^\eta(\frac{A^{d+1}-1}{A-1})^\eta \\ &= \frac{M'\nu^\eta A^\eta}{(A-1)^\eta}(A^d - \frac{1}{A})^\eta \\ &= \left(\frac{M'\nu^\eta A^\eta}{(A-1)^\eta}\right)(1-\frac{1}{A^{d+1}})^\eta A^{\eta d} \end{aligned}$$

Since for sufficiently large d, we know that $\frac{1}{2} \leq (1-\frac{1}{A^{d+1}})^\eta < 1$, the basic result holds. Shifting by $\pm\nu$n will not change this exponential rate of $A^{\eta d}$. □

Notice that Theorem 3.3.2 gives a lower bound to the sensitivity of all finite expected distortion source codes for the scalar Markov source, not just for causal ones. But using "non-causal" codes (ie. tolerating more end-to-end delay) can help us improve performance.

4.2 Performance in the limit of large delays

We now examine the question of optimal performance and focus on the squared error case for simplicity. The sequential rate-distortion theorems [63] bound the expected performance if we do not allow any delay. The natural extension would be to look at the scalar source as a vector source over blocks of inputs and then take a limit of the sequential rate-distortion as this block-length increases. This gives a bound, but it is not clear that it is achievable over even noiseless channels. After all, in general the sequential performance is not achievable without delay over a non-matched channel[63]. Alternatively, we could think of the infinite horizon problem as being the limit of a sequence of finite horizon problems as follows:

Definition 4.2.1 *Given a scalar Markov source from Definition 2.1.1, the N* finite horizon *version of that source is defined to be the random variables $X_1^N = (X_1, \ldots, X_N)$.*

For each N, we can calculate a standard information-theoretic rate-distortion function for the finite horizon problem as follows:

$$R_N^X(D) = \inf_{p_{\hat{X}_1^N|X_1^N} : \frac{1}{N}\sum_{i=1}^N E[\rho(X_i,\hat{X}_i)] \leq D} \frac{1}{N} I(X_1^N;\hat{X}_1^N) \tag{4.1}$$

In equation 4.1, we infimize the average mutual information between X and $\hat{X}$ over joint measures where the marginal for X_1^N is fixed and the average distortion $\rho(X_i,\hat{X}_i) = (X_i - \hat{X}_i)^2$ is constrained to be below D. We can think of the block X_1^N as a single vector-valued random variable $\vec{Y}$. Then, the $R_N^X(D)$ defined by equation 4.1 is very simply related to $R_1^{\vec{Y}}(D)$ by $R_N^X(D) = \frac{1}{N}R_1^{\vec{Y}}(ND)$ with the distortion measure on $\vec{Y}$ given by $\rho(\vec{Y},\hat{\vec{Y}}) = \|\vec{Y} - \hat{\vec{Y}}\|^2$. Thus, for the finite horizon problem sitting by itself, the standard data processing inequality shows us that $R_N^X(D)$ provides a lower bound on the average rate required by any source encoder/decoder system that achieves average distortion D or better.

It is easy to see that the infinite horizon average rate operationally required for a specified average distortion is lower bounded by the limit of the finite horizon case as follows:

$$R_{\infty}^{\text{oper}}(D) \geq R_{\infty}^{X}(D) = \liminf_{N\to\infty} R_N^X(D) \tag{4.2}$$

The question is whether this bound can be approached arbitrarily closely if we tolerate large end-to-end delays. For the finite horizon problem, the rate-distortion lower bound on average rate can be approached with a deterministic encoder if instead of one sample of $\vec{Y}$, we encode a block $\vec{Y}_1^M$ of M independent samples as M gets large. Mathematically, the classical rate-distortion source coding theorems[23] tell us that for every $R > R_T^X(d)$, there exists an $M > 0$ and a block source-code (encoder F_M and decoder G_M) such that:

$$E\left[\frac{1}{M}\sum_{i=1}^{M} \|\vec{Y}_i - \hat{\vec{Y}}_i\|^2\right] \leq ND$$

where $\hat{\vec{Y}}_1^M = G_M(F_M(\vec{Y}_1^M))$ and $\text{length}(F_M(\vec{Y}_1^M)) \leq MNR$.

We want to be able to somehow apply this well known result to show that the lower bound given by (4.2) is achievable in the limit of large end delays. The idea is to have the encoder use the delay available to it to process the source stream X_1^∞ into something which looks like an i.i.d. stream of $\vec{Y}_1^\infty$. This stream could then be encoded through a suitable large block encoder and then the decoder side could put the pieces $\hat{\vec{Y}}_1^\infty$ back together again to give a suitable reconstruction process $\hat{X}_1^\infty$.

4.3 False Starts: Why traditional approaches will not work

Before we give a technique which does work, it is worthwhile to consider a few ideas which do not work for the unstable case being considered. This will motivate the discussion in the next part as well as point out the connections to prior work.

4.3.1 "Whitening" Is Not Enough

The first observation which might occur to someone is that $(X_{t+n} - A^n X_t)$ has the same distribution as X_n. Furthermore, $(X_{t+n} - A^n X_t)$ and $(X_{t'+n} - A^n X_{t'})$ are independent as long as $|t - t'| \geq n$. So, the following mapping generates $\vec{Y}$ with the appropriate properties:

$$\left(\vec{Y}_i\right)_j = X_{iN+j} - A^j X_{iN} \tag{4.3}$$

where j goes from 1 to N and indexes the elements of the vector $\vec{Y}_i$. Within a given vector $\vec{Y}_i$, the elements are related recursively as:

$$\left(\vec{Y}_i\right)_{j+1} = A\left(\vec{Y}_i\right)_j + W_j$$

this recurrence holds all the way down to $j = 0$ an implicit element $\left(\vec{Y}_i\right)_0 = 0$. Since the underlying driving noise process W_k is i.i.d., the $\{\vec{Y}_i\}$ process is as well.

Notice also that the transformation is recursively invertible. $X_0 = 0$ by assumption and we can compute the rest by :

$$X_k = \left(\vec{Y_i}\right)_j + A^j X_{iN} \tag{4.4}$$

where $1 \le j \le N$ and $k = iN + j$.

One might presume that since the original transformation is invertible, we should be able to apply the inverse transformation to the compressed $\{\hat{Y}_i\}$ process to get a reasonable reconstruction process $\{\hat{X}_k\}$ for the original process. There is one obvious problem with approach. The inverse transformation given by (4.4) requires us to know X_{iN} exactly. Since this quantity is not available at the source decoder, we must modify (4.4) in some way. The natural thought is to use:

$$\hat{X}_k = \left(\hat{\vec{Y}}_i\right)_j + A^j \hat{X}_{Ni} \tag{4.5}$$

The problem with this approach is that the recursive inverse transformation specified by (4.5) is very sensitive and magnifies the errors:

$$\begin{aligned}
X_k - \hat{X}_k &= X_k - \left(\hat{\vec{Y}}_i\right)_j - A^j \hat{X}_{Ni} \\
&= X_k - \left(\vec{Y_i}\right)_j + (\left(\vec{Y_i}\right)_j - \left(\hat{\vec{Y}}_i\right)_j) - A^j \hat{X}_{Ni} \\
&= (\left(\vec{Y_i}\right)_j - \left(\hat{\vec{Y}}_i\right)_j) + X_k - (X_k - A^j X_{Ni}) - A^j \hat{X}_{Ni} \\
&= \left(\vec{Y_i}\right)_j - \left(\hat{\vec{Y}}_i\right)_j + A^j (X_{Ni} - \hat{X}_{Ni})
\end{aligned}$$

Thus, even if $E[\|\vec{Y_i} - \hat{\vec{Y}}_i\|] \le ND$ for all $i \ge 0$, the resulting $\{\hat{X}_k\}$ process will enjoy no such property. Rather, the errors will roughly get multiplied by A^N with every new block and hence the expected squared error will go to infinity. The problem is that the errors for different $\hat{Y}$ blocks will tend to be independent of each other and hence will not cancel out.

4.3.2 Predictive Error: Chicken and Egg Problem

Another choice would be to code a block version of the predictive error signals as is done in the proof of Theorem 4.1.1. Assume that we have access to the $\{\hat{X}_k\}$ process. Then we can generate the $\vec{Y_i}$ as follows:

$$\left(\vec{Y_i}\right)_j = X_{iN+j} - A^j \hat{X}_{iN} \tag{4.6}$$

where j goes from 1 to N and indexes the elements of the vector $\vec{Y_i}$ just as it did in (4.3). As before, within a given vector $\vec{Y_i}$, the elements are related recursively as:

$$\left(\vec{Y_i}\right)_{j+1} = A\left(\vec{Y_i}\right)_j + W_j$$

but this time, the recurrence is clearly valid only for $j \ge 1$. For $j = 0$, we have:

$$\begin{aligned}
\left(\vec{Y_i}\right)_1 &= X_{iN+1} - A\hat{X}_{iN} \\
&= AX_{iN} + W_{iN} - A\hat{X}_{iN} \\
&= A(X_{iN} - \hat{X}_{iN}) + W_{iM}
\end{aligned}$$

We can think of this as meaning that there is an implicit element $\left(\vec{Y}_i\right)_0 = X_{iN} - \hat{X}_{iN}$. This makes the recurrence valid for all $j \geq 0$, but introduces a non-zero initial condition. Furthermore, since this initial condition depends on X_{iN} (which also appears in the definition of the previous vector $\vec{Y}_{i-1}$) introduces a potential dependence between the successive $\vec{Y}_i$. It is doubtful even whether the $\{\vec{Y}_i\}$ process is stationary. Putting this difficulty aside for the moment, assume that we have some way of compressing a block of the $\{\vec{Y}_i\}$ and getting corresponding $\{\hat{\vec{Y}}_i\}$ which are close in the sense of having $E[\rho(\vec{Y}_i - \hat{\vec{Y}}_i)] \leq ND$ for all $i \geq 0$.

The real motivation of this approach is visible when we attempt to invert the transformation on the $\{\hat{\vec{Y}}_i\}$ process to get the $\{\hat{X}_k\}$. We take the appropriate analogue of (4.4) to get:

$$\hat{X}_k = \left(\hat{\vec{Y}}_i\right)_j + A^j \hat{X}_{iN}$$

where $1 \leq j \leq N$ and $k = iN + j$. Now, we can evaluate the error terms to see:

$$\begin{aligned}
X_k - \hat{X}_k &= X_k - \left(\hat{\vec{Y}}_i\right)_j - A^j \hat{X}_{iN} \\
&= X_k - \left(\vec{Y}_i\right)_j + (\left(\vec{Y}_i\right)_j - \left(\hat{\vec{Y}}_i\right)_j) - A^j \hat{X}_{iN} \\
&= (\left(\vec{Y}_i\right)_j - \left(\hat{\vec{Y}}_i\right)_j) + X_k - (X_k - A^j \hat{X}_{iN}) - A^j \hat{X}_{iN} \\
&= \left(\vec{Y}_i\right)_j - \left(\hat{\vec{Y}}_i\right)_j
\end{aligned}$$

Unlike the previous case, this time the errors do not propagate from block to block and under any difference distortion measure, our performance will be as good as that achieved by the $\{\hat{\vec{Y}}_i\}$. But this entire discussion overlooks one major problem: *In order to generate the $\{\vec{Y}_i\}$ process, we must have access to $\{\hat{X}_{iN}\}$. However, the $\{\hat{X}_k\}$ are functions of $\{\hat{\vec{Y}}_i\}$ which are themselves the compressed versions of $\{\vec{Y}_i\}$!* Because in general, the rate-distortion limit cannot be approached by purely causal encodings even of memoryless random variables[48, 9], at least some form of limited noncausality (like block encodings) is usually necessary. This leaves us with a "chicken and egg" problem: to get $\{\vec{Y}_i\}$, we need the $\{\hat{X}_k\}$ for which we need those very same $\{\vec{Y}_i\}$.

4.3.3 Berger's Technique

Toby Berger gave one way around this "chicken and egg" problem in his paper on "Information Rates of Wiener Processes"[5] and subsequently in his book on rate-distortion theory[6]. The system is described in Figure 4-2. The key observation is that to apply (4.5), we only need the prior X at the end of the last block of M samples. So, suppose that instead of using X_{iN}, we used a $\tilde{X}_{iN}$ which was guaranteed to always be close to X_{iN}. Then, we would have:

$$\hat{X}_k = \left(\hat{\vec{Y}}_i\right)_j + A^j \tilde{X}_{iN} \tag{4.7}$$

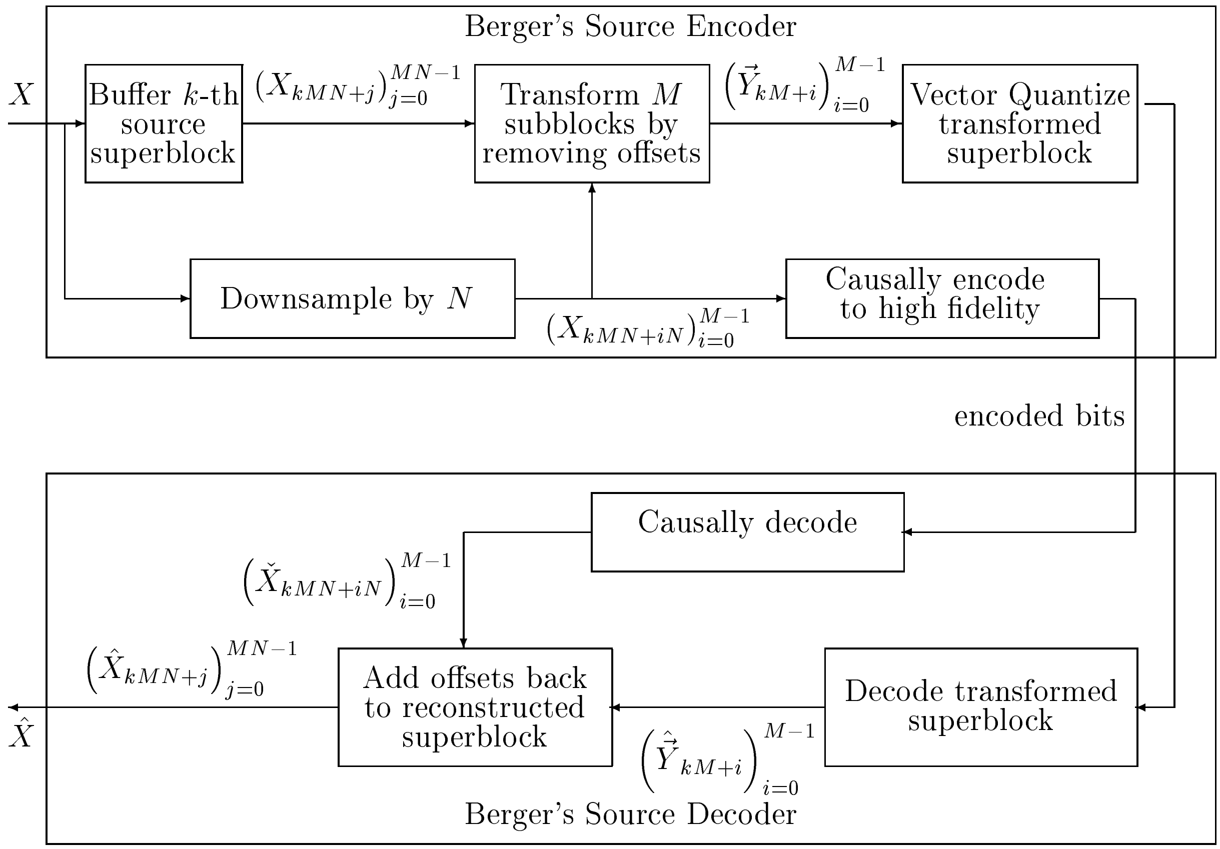

Figure 4-2: Berger's strategy achieving $R(D)$ for Wiener processes.

while the $\vec{Y}_i$ are generated by (4.3). It is easy to see that with this scheme:

$$X_k - \hat{X}_k = \left(\vec{Y}_i\right)_j - \left(\hat{\vec{Y}}_i\right)_j + A^j(X_{iN} - \tilde{X}_{iN})$$

In order to get $\tilde{X}_{iN}$, the idea is to causally encode in parallel the N-downsampled process consisting of: $(X_0, X_N, X_{2N}, \ldots, X_{iN}, \ldots)$ so that $|\tilde{X}_{iN} - X_{iN}| \leq \epsilon$. The hope is that the average added rate required to encode this parallel stream to fidelity ϵ can be made negligible relative to the rate required to encode the main stream of the $\{\vec{Y}_i\}$ to within expected distortion D, at least in the limit of large N.

The first question is what fidelity ϵ is required on the $\tilde{X}$ in order to be within a factor of $(1+\epsilon')$ of D on $E[(X_k - \hat{X}_k)^2]$ for all k. Clearly, this can be done if:

$$\max_{1 \leq j \leq N} |A^j| \epsilon < \frac{\epsilon'}{2}$$

For the cases that Berger considered where $|A| \leq 1$, the maximum is achieved at $j = 1$ and thus the choice of $\epsilon < \frac{\epsilon'}{2}$ has no overt dependence on N. But for the cases we are interested in, where $|A| \geq 1$, the maximum is achieved at $j = N$ and hence we need $\epsilon < \frac{\epsilon'}{2A^N}$.

The more important issue regards the $(X_0, X_N, X_{2N}, \ldots, X_{Ni}, \ldots)$ process:

$$X_{N(i+1)} = A^N X_{iN} + \tilde{W}_i \tag{4.8}$$

where the $\tilde{W}_i$ and are defined by:

$$\tilde{W}_i = \sum_{j=0}^{N-1} A^{N-1-j} W_{iN+j}$$

If the original W_i were i.i.d. so are the $\tilde{W}_i$ by inspection. Similarly, if the original W_i had finite support, then for any finite N, so do the $\tilde{W}_i$.

In the cases that Berger considers, with $|A| \leq 1$, the fixed rate required to causally encode $\tilde{X}_{iN}$ to a fidelity ϵ grows only sublinearly in N. For $|A| < 1$ this is obvious since with increasing N the $\{\tilde{X}_{iN}\}$ approach an i.i.d. sequence. For $|A| = 1$, Berger proves that the rate required is $O(\log N)$ and hence sublinear. For $|A| > 1$, we know from the work on sequential rate distortion[63], that encoding the $\tilde{X}_{iN}$ to any finite fidelity will require at least $\log_2(A^N) = N \log_2 A$ bits for every sample $\tilde{X}_{iN}$. This is linear in N even before we take into account our need for increasing accuracy as N gets larger! Thus Berger's approach strictly applied does not work for proving a source coding theorem for the exponentially unstable processes. This difficulty was implicitly recognized by Gray in his 1970 paper on "Information Rates of Autoregressive Processes"[29] where he comments:

> It should be emphasized that when the source is non-stationary, the above theorem is not as powerful as one would like. Specifically, it does not show that one can code a long sequence by breaking it up into smaller blocks of length n and use the same code to encode each block. The theorem is strictly a "one-shot" theorem unless the source is stationary, simply because the blocks $[(k-1)n, kn]$ do not have the same distribution for unequal k when the source is not stationary.
>
> Berger has proved the stronger form of the positive coding theorem for the Wiener process by using a specific coding scheme (delta modulation) to track the starting point of each block.

Gray and other subsequent researchers were not able to generalize Berger's technique to the general unstable case because as our sequential rate-distortion based insight shows, the purely parallel streams argument cannot work for the exponentially unstable case.

4.4 Variable Rate Technique

In this section, we show how it is possible to use a combination of ideas to give a variable length source code which allows us to approach the bound in (4.2) arbitrarily closely as we let the end-to-end delay get large. Our basic approach will be to assume that both the encoder and decoder have access to common randomness, and then to use that randomness to make a modified version of Berger's technique work with an appropriately dithered version of the predictive error based transform.

A variable length source code is where the encoder is allowed at each time step to produce symbols from $\{0,1\}^* = \bigcup_{l=1}^{\infty} \{0,1\}^l$ which we can identify with all the positive integers rather than just a finite subset of them.

Figure 4-3 summarizes how the encoders and decoders will work. Here is a summary of the encoding procedure:

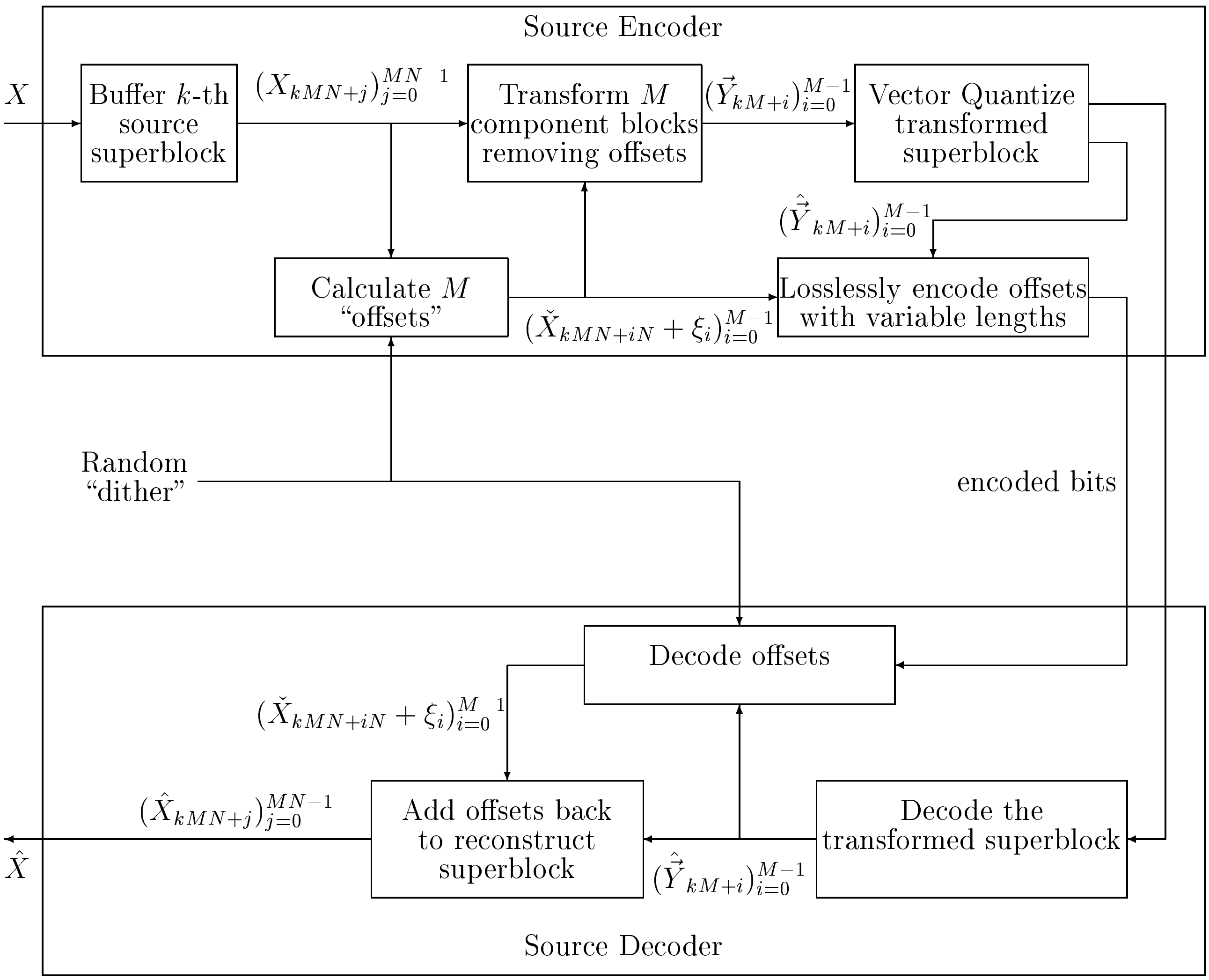

Figure 4-3: The two-part superblock source encoding strategy for Markov sources and how it is decoded.

1. The incoming source will be buffered into blocks of length MN. These will be thought of as superblocks consisting of M blocks each containing N consecutive source samples.

2. We will quantize the endpoint of the i-th component block to a uniform grid with spacing θ between representative points. This grid will be shifted for each i by an i.i.d. uniform random dither signal of width θ. Furthermore, an additional i.i.d. random variable ξ_i is added to the representative points. All this randomness is assumed to be common randomness accessible to the decoder as well. (This is a slight simplification of what goes on. In section 4.4.2 we will give the whole story which includes some added technical manipulations.)

3. The superblock from step 1 is transformed by appropriately translating each component block down using the offsets calculated in step 2. (Depicted in Figure 4-4) This is analogous to the way in which the predictions are subtracted off in equation (4.6).

4. The order of the constituent blocks is shuffled (randomly permuted) in the transformed superblock from step 3 using common randomness. (Not depicted in Figure 4-3 or 4-4, but used for technical purposes in the proof.)

5. The shuffled transformed superblock from step 4 is treated as a block of M samples from an i.i.d. source and is quantized using a good vector quantizer.

6. *Key Step:* The positions of the quantized endpoints from step 2 (used to do the transformations in step 3) are losslessly encoded using a recursive variable-rate code relative to the $\hat{X}_i$ induced by the vector quantization done in step 5. Lossless encoding is possible because conditioned on the common random "dither" signals, the quantized endpoints must lie on a countable set. Doing this step resolves the "chicken and egg" dilemma.

7. The fixed-rate bits coming from step 5 and the variable-rate ones from step 6 are sent across to the decoder.

The operation of the decoder in Figure 4-3 has access to all the common randomness and the encoded bitstream. It reverses the procedure as follows:

1. The bits are buffered up until we have enough to decode the vector quantized transformed superblock from the encoder's Step 5.

2. Using the common randomness used in the encoder's Step 4, the order of the constituent blocks is deshuffled (using the inverse permutation). (This technical step is not depicted in Figure 4-3)

3. *Key Step:* The bits from the variable length codes are recursively combined with past information and the constituent blocks emerging from the decoder's Step 2 to reconstruct the quantized endpoints of encoder Step 2.

4. The recovered endpoints from the decoder's Step 3 are combined with the recovered quantized constituent blocks from decoder's Step 2 to give us estimates $\{\hat{X}_i\}$ for the original process.

Now, we will analyze the performance of the coding strategy given in Figure 4-3 in order to see that it works asymptotically.

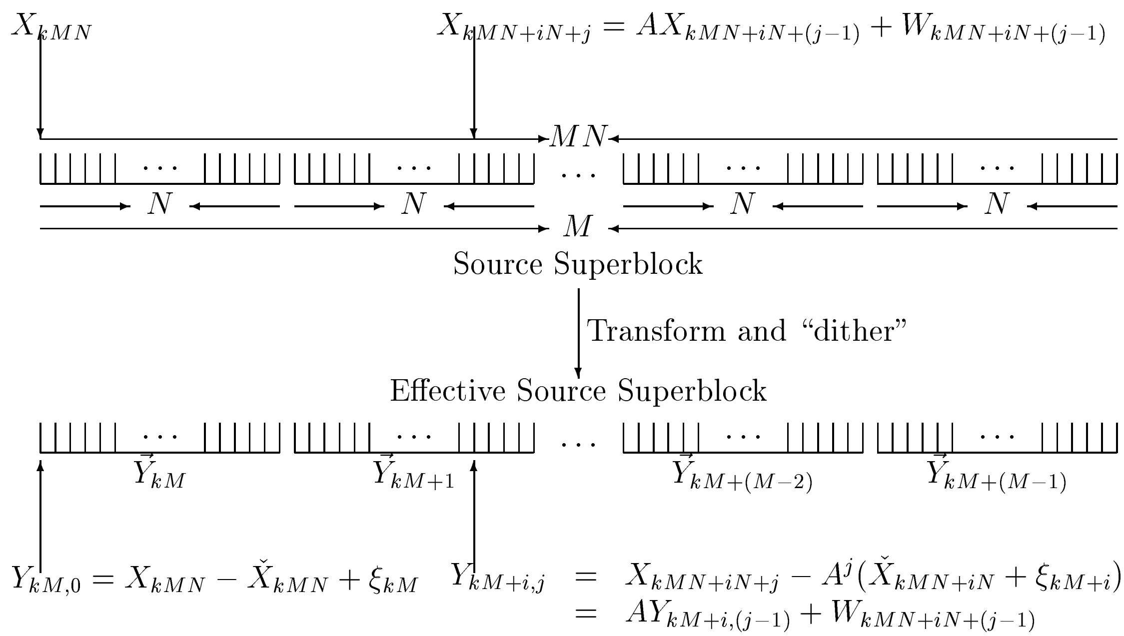

Figure 4-4: The "block of blocks" approach with a long superblock consisting of M inner blocks of N source samples each. The transformed inner blocks $\vec{Y}_{kM+i}$ all look identically distributed and independent of each other so that they are suitable for lossy compression via vector quantization.

4.4.1 The key transformation

The key transformation is the one given by the Encoder's Step 3 and is depicted in Figure 4-4. It generates the vector $\vec{Y}_{kM+i}$ from the i component block of the k-th superblock and the value for $\check{X}_{kMN+iN} + \xi_{kM+i}$ generated by the encoder's Step 2 as follows (for $j \geq 1$:

$$\begin{aligned} Y_{kM+i,j} &= X_{kMN+iN+j} - A^j(\check{X}_{kMN+iN} + \xi_{kM+i}) \\ &= AY_{kM+i,(j-1)} + W_{kMN+iN+(j-1)} \end{aligned} \tag{4.9}$$

where the first term ($j = 0$) is given by:

$$Y_{kM+i,0} = X_{kMN+iN} - \check{X}_{kMN_iN} + \xi_{kM+i}$$

If we add an implicit zero initial condition at position -1 in this vector, we notice that the distribution of the N vector $\vec{Y}_{kM+i}$ is almost identical to that of the N finite horizon version of the original source. The only difference is that rather than having something which looks like $W_{kMN+iN-1}$ at the first position, it has $(X_{kMN+iN} - \check{X}_{kMN_iN}) + \xi_{kM+i}$. In section 4.4.2, we will show how under appropriate conditions, we can actually make this term be independent of all the other $\{W_j\}$ and share the same distribution as them. (Captured as Property 4.4.1) Once that is true, the $\vec{Y}_{kM+i}$ are all i.i.d. and share the same distribution as the N finite horizon version of the original source.

In this memoryless setup, we know by classical information theory[23] that given an M large enough, we can construct a vector encoder so for a given distortion D, we are within ϵ_1 of the rate-distortion function. Once we have done the technical random shuffling of Encoder Step 4, we have the following Lemma:

Lemma 4.4.1 *Assume we have an i.i.d. sequence of random variables $\vec{Y}_1^M$ with information-theoretic rate distortion function $R_1^{\vec{Y}}(D)$ under additive distortion measure ρ. Then for every $\epsilon > 0$, there exists an $M > 0$ and shuffling block source code of size M which achieves $E\left[\rho(\vec{Y}_i, \hat{\vec{Y}}_i)\right] \leq D$ for every i with total rate less than or equal to $M(R_1^{\vec{Y}}(D) + \epsilon)$.*

Proof: This is only a minor variation of the standard result in information theory. The standard result tells us that for every $\epsilon > 0$, there exists an $M > 0$ and deterministic block source code of size M which achieves $E\left[\frac{1}{M}\sum_{i=1}^{M}\rho(\vec{Y}_i, \hat{\vec{Y}}_i)\right] \leq D$ with total rate less than or equal to $M(R_1^{\vec{Y}}(D) + \epsilon)$.

The shuffling permutation, since it happens after the choice of the encoder, tells us that the expected behavior of every position must be the same. So since $E\left[\rho(\vec{Y}_i, \hat{\vec{Y}}_i)\right] = E\left[\rho(\vec{Y}_j, \hat{\vec{Y}}_j)\right]$, we can conclude that:

$$\begin{aligned} D &\geq E\left[\frac{1}{M}\sum_{i=1}^{M}\rho(\vec{Y}_i, \hat{\vec{Y}}_i)\right] \\ &= \frac{1}{M}\sum_{i=1}^{M} E\left[\rho(\vec{Y}_i, \hat{\vec{Y}}_i)\right] \\ &= E\left[\rho(\vec{Y}_j, \hat{\vec{Y}}_j)\right] \end{aligned}$$

For every position j. $\square$

For now, assume that we somehow know the sequence of $\check{X}_{kMN+iN} + \xi_{kM+i}$ exactly at the decoder. (This will be treated in greater detail in Section 4.4.3) Then, we can put together an estimate of the original $\{X_j\}$ process by the following procedure:

$$\hat{X}_{kMN+iN+j} = \hat{Y}_{kM+i,j} + A^j(\check{X}_{kMN+iN} + \xi_{kM+i}) \tag{4.10}$$

We can easily evaluate the performance under any difference distortion measure (such as squared error or η-error) by noticing that:

$$\begin{aligned} \hat{X}_{kM+iN+j} - X_{kM+iN+j} &= (\hat{Y}_{kM+i,j} + A^j(\check{X}_{kMN+iN} + \xi_{kM+i})) \\ &\quad -(Y_{kM+i,j} + A^j(\check{X}_{kMN+iN} + \xi_{kM+i})) \\ &= \hat{Y}_{kM+i,j} - Y_{kM+i,j} \end{aligned}$$

There is no deadly compounding of errors in this decoding system as long as we know the $\check{X}_{kMN+iN} + \xi_{kM+i}$ exactly.

So, by the construction of the vector quantizer and Lemma 4.4.1 we know that for every block $X_{kMN+iN}^{kMN+iN+(N-1)}$, we have:

$$E\left[\frac{1}{N}\sum_{j=0}^{N-1}(X_{kMN+iN+j} - \hat{X}_{kMN+iN+j})^\eta\right] \leq D$$

And hence the same thing holds for the limiting average distortion.

Now, assume that we do not know $\check{X}_{kMN+iN} + \xi_{kM+i}$ exactly and instead only know them to some fidelity θ'. So at the decoder, we only have access to $\check{X}_{kMN+iN} + \xi_{kM+i} + Q_{kM+i}$ where $|Q_{kM+i}| \leq \theta'$ but has zero mean. Then it is immediate that we will be forced to use:

$$\hat{X}_{kM+iN+j} = \hat{Y}_{kM+i,j} + A^j(\check{X}_{kMN+iN} + \xi_{kM+i} + Q_{kM+i}) \tag{4.11}$$

and will get mean-squared error performance:

$$\begin{aligned}
& E\left[\frac{1}{N}\sum_{j=0}^{N-1}(X_{kMN+iN+j} - \hat{X}_{kMN+iN+j})^2\right] \\
= \quad & E\left[\frac{1}{N}\sum_{j=0}^{N-1}(Y_{kM+i,j} - \hat{Y}_{kM+i,j} - A^j Q_{kM+i})^2\right] \\
= \quad & E\left[\frac{1}{N}\sum_{j=0}^{N-1}(Y_{kM+i,j} - \hat{Y}_{kM+i,j})^2 + A^{2j}(Q_{kM+i})^2 + 2(Y_{kM+i,j} - \hat{Y}_{kM+i,j})Q_{kM+i}\right] \\
\leq \quad & D + \frac{1}{N}\sum_{j=0}^{N-1} A^{2j}\theta'^2 \\
= \quad & D + \frac{A^{2N}-1}{N(A^2-1)}\theta'^2
\end{aligned}$$

4.4.2 The full story of the "dithering"

In the past section, we had to assume one major property for $\check{X}_{kMN+iN} + \xi_{kM+i}$:

Property 4.4.1 $(X_{kMN+iN} - \check{X}_{kMN_iN} + \xi_{kM+i})$ *has the same distribution as* W_j *and is independent of all the* W_j *in the original system as well as* $(X_{k'MN+i'N} - \check{X}_{k'MN_i'N} + \xi_{k'M+i'})$ *for* $(k', i') \neq (k, i)$

Before we establish that this property holds, let us first recall how it is that we are going to generate the $\check{X}_j$ in the encoder's Step 2.

1. A common random i.i.d. process of uniform random variables $\{\Psi_t\}$ is sampled at j.
2. We quantize $X_t + \Psi_t$ to an appropriate point $M_t\theta$ on the uniform grid of spacing θ. In particular, we generate the integer M_t by:

$$M_t = \left\lfloor \frac{X_t + \Psi_t}{\theta} \right\rfloor \tag{4.12}$$

3. Let $\check{X}_t = M_t\theta - \Psi_t$

Lemma C.2.2 tells us immediately that $\Phi_t = (X_t - \check{X}_t) = X_t - M_t\theta + \Psi_t$ is an i.i.d. uniform random variable. Once we have this property, we know that by appropriate choice of common random i.i.d. process $\{\xi_t\}$ we can make $(X_{kMN+iN} - \check{X}_{kMN_iN} + \xi_{kM+i})$ have Property 4.4.1 as long as the distribution for the W_j is additively constructible out of uniform random variables of appropriate width θ. (See Appendix C for a discussion of this and other notions of constructibility.) If W_j is not additively constructible out of uniform random variables, then all we can hope to get by this procedure is a close approximation to W_j and hence our $\vec{Y}$ will have a distribution which is only an approximation to X_1^N rather than having the exact same distribution.

Although it is possible to continue with this proof and to deal with the resulting approximation error explicitly (see Section 4.5.3), at the moment we will follow a different strategy and augment our procedure slightly:

1. Flip a common random biased i.i.d. coin B_t with $P(1) = \epsilon_2$

2. If $B_t = 0$, stop.

3. Otherwise, sample another common random i.i.d. random variable Z_t

4. Discard the old value for $\check{X}_t$ and set $\check{X}_t + \xi_t = X_t + Z_t$ instead.

Under this procedure, by setting ϵ_2 and picking the distribution for Z_t appropriately, we can satisfy Property 4.4.1 as long as W_t has a distribution which is ϵ_2-approximately additively constructible out of uniform random variables of appropriate width θ. So, we have proved the following:

Proposition 4.4.1 *If the probability distribution for W_t is ϵ_2-approximately additively constructible out of uniform random variables with appropriate width θ, then we can choose (θ, ξ_t, Z_t) so that using our procedure satisfies Property 4.4.1.*

4.4.3 Encoding the offsets

The important insight which distinguishes this approach from a straightforward application of Berger's technique is in the way that the offset stream $\check{X}_{kMN+iN} + \xi_{kM+i}$ is encoded. As Figure 4-3 shows, rather than encoding the offset stream into bits independently, we will use the $\{\hat{\vec{Y}}\}$ to reduce the number of bits required for this purpose. The important thing that we need to do is to make sure that the encoding can be decoded by the decoder with access only to information known at the decoder. $\{\hat{\vec{Y}}\}$ can be assumed to be available at the decoder since decoding the source code for $\vec{Y}$ in Lemma 4.4.1 only needs access to the common randomness of the shuffling and the encoded bits themselves.

The random variables $(B_t, \xi_t, Z_t, \Psi_t)$ are all also presumed to be "common randomness" and known to the decoder. First, let us assume that $B_t = 0$. In this case, $\check{X}_t + \xi_t = M_t\theta - \Psi_t + \xi_t$ is what we want to encode, and the only part of that which is presumed not to be known to the decoder is the $M_t\theta$ part. The encoding will proceed by induction. Since the initial condition for the original $\{X_t\}$ process is fixed at 0, the base case is not an issue. Assume that we know $\check{X}_{kMN+(i-1)} + \xi_{kM+i-1}$. Given that we know $\hat{Y}_{kM+i,N-1}$, using (4.11), we can compute $\hat{X}_{kMN+(i-1)N+N-1} = \hat{X}_{kMN+iN-1}$. We can now calculate the following analog of Equation (4.12):

$$\hat{M}_{kMN+iN} = \left\lfloor \frac{A\hat{X}_{kMN+iN-1} + \Psi_{kMN+iN-1}}{\theta} \right\rfloor \tag{4.13}$$

As Figure 4-5 shows, the idea is to encode the integer M_{kMN+iN} by losslessly encoding $(M_{kMN+iN} - \hat{M}_{kMN+iN})$ using some self-punctuating variable length code on the integers. Such codes (eg. the code which uses one bit for the sign and then encodes the magnitude using the standard binary representation but with a '1' inserted after every bit except the least significant bit which is followed by a '0') assign longer codewords to integers with

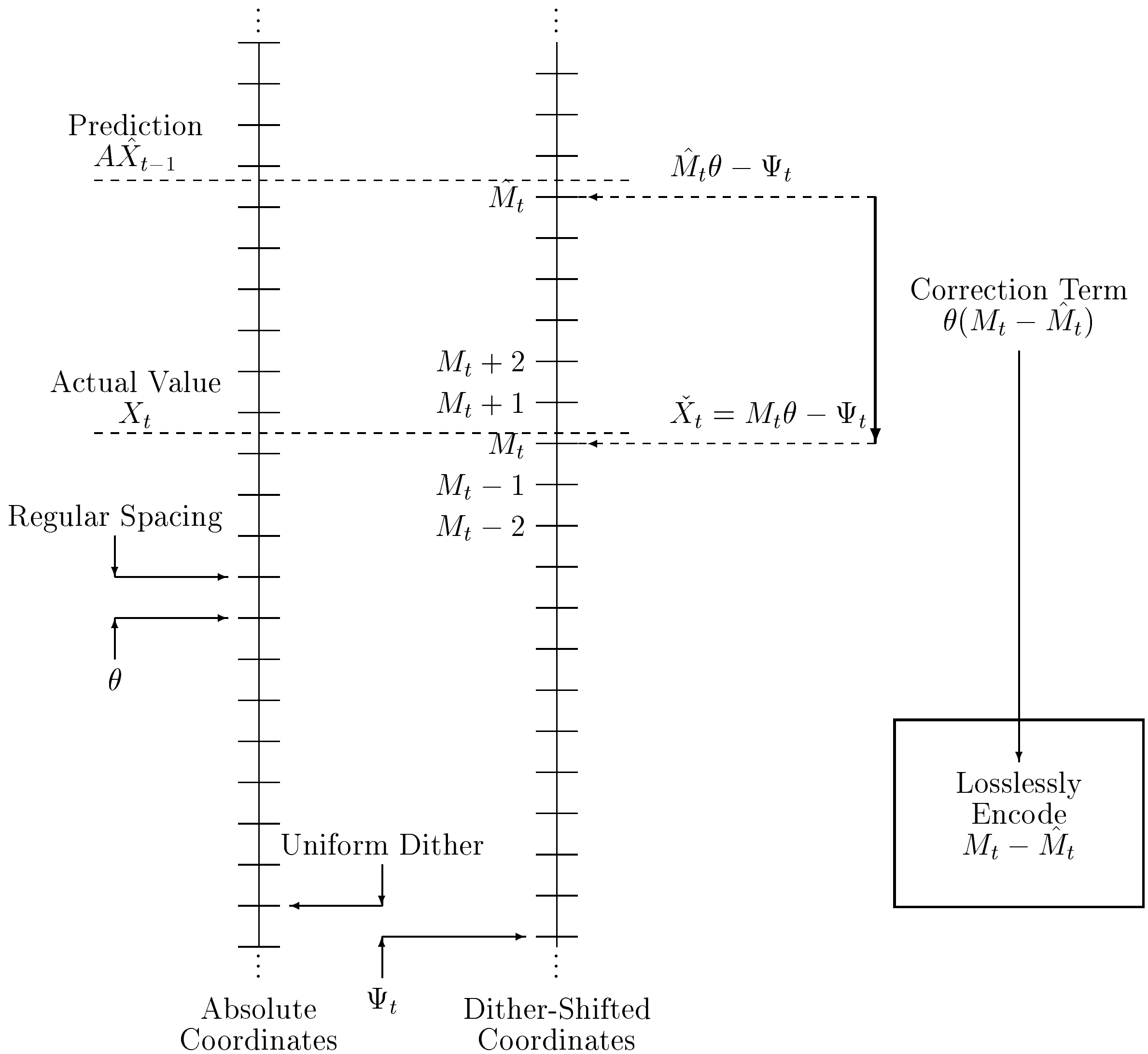

Figure 4-5: How the dithered quantization feeds into the variable length lossless code

larger magnitudes in a logarithmic fashion. So we need to understand the probability that the magnitude is large by looking at:

$$
\begin{aligned}
|M_t - \hat{M}_t| &= |\left\lfloor \frac{X_t + \Psi_t}{\theta} \right\rfloor - \left\lfloor \frac{A\hat{X}_{t-1} + \Psi_t}{\theta} \right\rfloor| \\
&= |\left\lfloor \frac{X_t + \Psi_t}{\theta} \right\rfloor - \left\lfloor \frac{\hat{X}_t - W_{t-1} + \Psi_t}{\theta} \right\rfloor| \\
&\leq 1 + |\frac{X_t + \Psi_t}{\theta} - \frac{\hat{X}_t - W_{t-1} + \Psi_t}{\theta}| \\
&= 1 + \frac{|(X_t - \hat{X}_t) + W_{t-1}|}{\theta} \\
&= 1 + \frac{|(X_t - \hat{X}_t)|}{\theta} + \frac{|W_{t-1}|}{\theta}
\end{aligned}
$$

By getting a bound on the probability of each of those terms, we are fine. For the first term, we have:

$$
\begin{aligned}
P(\frac{|(X_t - \hat{X}_t)|}{\theta} \geq O) &= P(|(X_t - \hat{X}_t)| \geq O\theta) \\
&\leq \min(1, \frac{E\left[(X_t - \hat{X}_t)^2\right]}{(O\theta)^2}) \\
&\leq \min(1, \frac{ND\theta^{-2}}{O^2})
\end{aligned}
$$

where the first inequality follows from appropriate version of Chebychev's inequality and the second comes from the fact that the expected distortion on the N-th component of the block can not exceed the expected total distortion on the entire block.

As long as W_t has a finite variance, Chebychev's inequality will give us an analogous bound:

$$
P(\frac{|W_{t-1}|}{\theta} \geq O) \leq \min(1, \frac{E\left[W_t^2\right]}{(O\theta)^2})
$$

Putting it all together, have:

$$
\begin{aligned}
P(|M_t - \hat{M}_t| \geq O) &\leq P(1 + \frac{|(X_t - \hat{X}_t)|}{\theta} + \frac{|W_{t-1}|}{\theta} \geq O) \\
&= P(\frac{|(X_t - \hat{X}_t)|}{\theta} + \frac{|W_{t-1}|}{\theta} \geq O - 1) \\
&\leq \max(P(\frac{|(X_t - \hat{X}_t)|}{\theta} \geq \frac{O-1}{2}), P(\frac{|W_{t-1}|}{\theta} \geq \frac{O-1}{2})) \\
&\leq \max(P(\frac{|(X_t - \hat{X}_t)|}{\theta} \geq \frac{O-1}{2}), P(\frac{|W_{t-1}|}{\theta} \geq \frac{O-1}{2})) \\
&= \min(1, \max\left(\frac{ND}{(\frac{O-1}{2}\theta)^2}, \frac{E\left[W_t^2\right]}{(\frac{O-1}{2}\theta)^2}\right)) \\
&= \min(1, 4\max(ND, E\left[W_t^2\right])((O-1)\theta)^{-2})
\end{aligned}
$$

Now, we can evaluate the tail probability for the length L_t of our lossless encoding of any particular M_t. We are interested in large lengths λ or bigger and so will assume that $|M_t - \hat{M}_t| > 2$ throughout. In that case, the length of the lossless code for integer $(M - \hat{M})$ can be upper bounded by $2 + (1 + 2\log_2(|M - \hat{M}| - 1))$ since an initial 2 bits are enough to specify whether we want to move up by 1, down by 1, or stay at the same place.

$$
\begin{aligned}
P(L_t \geq \lambda) &\leq P(3 + 2\lceil \log_2(|M_t - \hat{M}_t| - 1)|\rceil \geq \lambda) \\
&\leq P(2\log_2(|M_t - \hat{M}_t| - 1) \geq \lambda - 3) \\
&\leq P(|M_t - \hat{M}_t| \geq 2^{\frac{\lambda-3}{2}} + 1) \\
&\leq \min(1, 4\max(ND^2, E\left[W_t^2\right])((2^{\frac{\lambda-3}{2}} + 1 - 1)\theta)^{-2}) \\
&= \min(1, 4\max(ND^2, E\left[W_t^2\right])\theta^{-2}2^{-\lambda+3}) \\
&= \min(1, \frac{32\max(ND^2, E\left[W_t^2\right])}{\theta^2}2^{-\lambda})
\end{aligned}
$$

Finally, we can calculate the expected length of this variable code:

$$
\begin{aligned}
E[L_t] &= \int_0^\infty P(L_t \geq \lambda)d\lambda \\
&\leq \int_0^\infty \min(1, \frac{32\max(ND^2, E\left[W_t^2\right])}{\theta^2}2^{-\lambda})d\lambda \\
&= \int_0^{5+2\log_2\left(\frac{1}{\theta}\right)+\max(\log_2(ND^2),\log_2(E\left[W_t^2\right]))} \min(1, \frac{32\max(ND^2, E\left[W_t^2\right])}{\theta^2}2^{-\lambda})d\lambda \\
&\quad + \int_{5+2\log_2\left(\frac{1}{\theta}\right)+\max(\log_2(ND^2),\log_2(E\left[W_t^2\right]))}^\infty \min(1, \frac{32\max(ND^2, E\left[W_t^2\right])}{\theta^2}2^{-\lambda})d\lambda \\
&= 5 + 2\log_2\left(\frac{1}{\theta}\right) + \max(\log_2 N + 2\log_2 D), \log_2(E\left[W_t^2\right])) + \int_0^\infty 2^{-\lambda}d\lambda \\
&= \frac{1}{\ln 2} + 5 + 2\log_2\left(\frac{1}{\theta}\right) + \max(\log_2 N + 2\log_2 D, \log_2(E\left[W_t^2\right]))
\end{aligned}
$$

So we have proved the following proposition:

Proposition 4.4.2 *Consider W_t with finite variance and approximately additively constructible out of uniform random variables of width θ. If $B_t = 0$ the length L_t of the encoding for $\check{X}_t + \xi_t$ under our variable length encoding procedure has*

$$
E[L_t] \leq \frac{1}{\ln 2} + 5 + 2\log_2\left(\frac{1}{\theta}\right) + \max(\log_2 N + 2\log_2 D, \log_2(E\left[W_t^2\right]))
$$

and a probability for large deviations that drops at least exponentially:

$$
P(L_t \geq \lambda) \leq \min(1, \frac{32\max(ND^2, E\left[W_t^2\right])}{\theta^2}2^{-\lambda})
$$

The important fact is that this length grows only logarithmically in N and $\frac{1}{\theta}$.

It remains now to consider the case where $B_t = 1$ which occurs with a probability ϵ_2. In this case, we need to encode an approximation to $\check{X}_{kMN+iN} + \xi_{kM+i}$ that gets us within an appropriately chosen θ' of its true value. $\check{X}_{kMN+iN} + \xi_{kM+i} = X_{kMN+iN} + Z_{kMN+iN}$ and Z_{kMN+iN} is presumably known because it is common randomness, all we have to do is encode X_{kMN+iN} to within an accuracy θ'. It is clear that this can be done using some more common randomness by a procedure exactly analogous to what we used in the case of $B_t = 0$ except using θ' instead of θ. Basically, we quantize X_{kMN+iN} to accuracy θ' and then losslessly encode the differential to it from the existing $\hat{X}_{kMN+iN-1}$. The only question which remains is how to choose θ'.

We know that the reconstruction will have:

$$E\left[\frac{1}{N}\sum_{j=0}^{N-1}(X_{kMN+iN+j} - \hat{X}_{kMN+iN+j})^2\right] \leq D + \frac{A^{2N}-1}{N(A^2-1)}\theta'^2$$

To guarantee that we get within ϵ_3 of D we need to choose θ' so that: $\theta' \leq \sqrt{\left(\frac{N(A^2-1)}{A^{2N}-1}\right)\epsilon_3}$. It is easy to see that

$$\theta' = \sqrt{N(A^2-1)\epsilon_3}A^{-N}$$

satisfies the inequality and gives the desired performance. Using this θ' in Proposition 4.4.2 gives us:

$$\begin{aligned}
E[L_t] &\leq \frac{1}{\ln 2} + 5 + \max(\log_2 N + 2\log_2 D, \log_2(E\left[W_t^2\right])) + 2\log_2\left(\frac{1}{\theta'}\right) \\
&= \frac{1}{\ln 2} + 5 + \max(\log_2 N + 2\log_2 D, \log_2(E\left[W_t^2\right])) + 2\log_2\left((N(A^2-1)\epsilon_3)^{-\frac{1}{2}}A^N\right) \\
&= \frac{1}{\ln 2} + 5 + \max(\log_2 N + 2\log_2 D, \log_2(E\left[W_t^2\right])) \\
&\quad - \log_2 N - \log_2(A^2-1) - \log_2\epsilon_3 + N\log_2 A
\end{aligned}$$

In this case, the expected length of the encoding has a linear term in the inner block length N. However, it is easy to see that the probability of large deviations in encoding length still drops exponentially at least as fast as $O(2^{-\lambda})$:

$$\begin{aligned}
P(L_t \geq \lambda) &\leq \min(1, \frac{32\max(ND^2, E\left[W_t^2\right])}{\theta'^2}2^{-\lambda}) \\
&= \min(1, \frac{32\max(ND^2, E\left[W_t^2\right])}{N(A^2-1)\epsilon_3 A^{-2N}}2^{-\lambda}) \\
&= \min(1, \frac{32\max(ND^2, E\left[W_t^2\right])}{N(A^2-1)\epsilon_3}A^{2N}2^{-\lambda})
\end{aligned}$$

Combining these results with those from Proposition 4.4.2 gives us:

Proposition 4.4.3 *Consider W_t with finite variance and approximately additively constructible out of uniform random variables of width θ with ϵ_2 probability left over. Then, the length L_t of the encoding for $\check{X}_t + \xi_t$ under our variable length encoding procedure has :*

$$E[L_t] < \frac{1}{\ln 2} + 5 + \log_2(E\left[W_t^2\right]) + \log_2 N + 2\log_2 D + 2\log_2\left(\frac{1}{\theta}\right) + \epsilon_2\left(\log_2\left(\frac{1}{\epsilon_3}\right) + N\log_2 A\right)$$

and furthermore, the probability for large deviations depends on B_t and drops as follows:

$$P(L_t \geq \lambda) \leq P(\bar{L}_t \geq \lambda)$$

where:

$$\begin{aligned}\bar{L}_t &= 5 + 2\log_2\left(\frac{1}{\theta}\right) + \log_2 N + 2\log_2 D + \log_2(E\left[W_t^2\right]) \\ &+ \left(\log_2\left(\frac{1}{\epsilon_3}\right) + N\log_2 A\right) B_t \\ &+ \tilde{L}_t\end{aligned}$$

and $\tilde{L}_t$ has exponential distribution: $P(\tilde{L}_t \geq \lambda) = 2^{-\lambda}$

Proof: As far as expected length goes:

$$\begin{aligned}E[L_t] &= (1-\epsilon_2)E[L_t|B_t=0] + \epsilon_2 E[L_t|B_t=1] \\ &\leq (1-\epsilon_2)\left(\frac{1}{\ln 2} + 5 + \max(\log_2 N + 2\log_2 D, \log_2(E\left[W_t^2\right])) + 2\log_2\left(\frac{1}{\theta}\right)\right) \\ &\quad +\epsilon_2\left(\frac{1}{\ln 2} + 5 + \max(\log_2 N + 2\log_2 D, \log_2(E\left[W_t^2\right])) + N\log_2 A\right) \\ &\quad -\epsilon_2\left(\log_2 N + \log_2(A^2-1) + \log_2\epsilon_3\right) \\ &< \frac{1}{\ln 2} + 5 + \log_2(E\left[W_t^2\right]) + \log_2 N + 2\log_2 D + 2\log_2\left(\frac{1}{\theta}\right) \\ &\quad +\epsilon_2\left(\log_2\left(\frac{1}{\epsilon_3}\right) + N\log_2 A\right)\end{aligned}$$

To see the result for the probability of large deviations, we notice that in the case that $B_t = 0$, the inequality $P(L_t \geq \lambda) \leq \min(1, \frac{32\max(ND^2, E\left[W_t^2\right])}{\theta^2}2^{-\lambda})$ can also be interpreted as saying that L_t can be bounded above by a simpler random variable $\bar{L}_t = 5 + 2\log_2\left(\frac{1}{\theta}\right) + \log_2 N + 2\log_2 D + \log_2(E\left[W_t^2\right]) + \tilde{L}_t$ where $\tilde{L}_t$ is a positive random variable with an exponential distribution $P(\tilde{L}_t \geq \lambda) = 2^{-\lambda}$. For the case $B_t = 1$, the same argument tells us that $\bar{L}_t = 5 + 2\log_2\left(\frac{1}{\theta}\right) + \log_2 N + 2\log_2 D + \log_2(E\left[W_t^2\right]) + \left(\log_2\left(\frac{1}{\epsilon_3}\right) + N\log_2 A\right) + \tilde{L}_t$. Combining them, we can bound L_t from above by:

$$\bar{L}_t = 5 + 2\log_2\left(\frac{1}{\theta}\right) + \log_2 N + 2\log_2 D + \log_2(E\left[W_t^2\right]) + \left(\log_2\left(\frac{1}{\epsilon_3}\right) + N\log_2 A\right) B_t + \tilde{L}_t$$

which proves the desired result. □

4.4.4 Putting the pieces together

We are now ready to prove the main source-coding theorem of this section:

Theorem 4.4.1 *If $\{X_t\}$ is a scalar Markov process with $A > 1$ driven by i.i.d. $\{W_t\}$ where the distribution for W_t is Riemann-Integrable on a bounded set and is continuous outside that set, then the information theoretic infinite horizon rate-distortion function $R_\infty^X(D)$ can be approached operationally by variable length codes with access to common randomness. That is, for every $\epsilon > 0$, there exists a commonly randomized variable length source code with finite end-to-end delay for which:*

$$\lim_{t\to\infty} E\left[\frac{1}{t}Length(\hat{X}_1^t)\right] \leq R_\infty^X(D) + \epsilon$$

while achieving:

$$\lim_{t\to\infty} E\left[\frac{1}{t}\sum_{k=1}^{t}(\hat{X}_k - X_k)^2\right] \leq D + \epsilon$$

Proof: All the pieces are in place. All we need to do is to show that it is possible to choose $(N, M, \theta, \epsilon_3)$ in order to achieve the desired performance. The fixed-rate part of the code is classical and any difficulty would arise from the variable-rate part. Recall that our variable rate procedure needs to encode $\check{X}_t + \xi_t$ only every N samples. Its contribution to the overall average rate, by Proposition 4.4.3, is:

$$\begin{aligned}
& \frac{E[L_t]}{N} \\
< \; & \frac{\frac{1}{\ln 2} + 5 + \log_2(E\left[W_t^2\right]) + \log_2 N + 2\log_2 D + 2\log_2\left(\frac{1}{\theta}\right) + \epsilon_2\left(\log_2\left(\frac{1}{\epsilon_3}\right) + N\log_2 A\right)}{N} \\
= \; & \left(\frac{\frac{1}{\ln 2} + 5 + \log_2(E\left[W_t^2\right]) + 2\log_2 D}{N}\right) \\
& + \left(\frac{\log_2 N}{N}\right) \\
& + \left(\frac{2\log_2\left(\frac{1}{\theta}\right) + \epsilon_2\log_2\left(\frac{1}{\epsilon_3}\right)}{N}\right) \\
& + \epsilon_2 \log_2 A
\end{aligned}$$

The first two terms in the sum can obviously be made arbitrarily small by choosing N large enough. The third term can also be made arbitrarily small as long as the choice of θ, ϵ_2, and ϵ_3 does not depend on N. This is true since θ and ϵ_2 are only related to the distribution for W_t and ϵ_3 was free for us to choose. The final term involves making ϵ_2 (the probability that $B_t = 1$) arbitrarily small, which is possible since W_t is arbitrarily additively constructible out of uniform random variables of width θ by Theorem C.2.1.

Making things explicit, given an ϵ, consider $\epsilon' = \frac{\epsilon}{6}$. Since the distribution for W_t is assumed to be arbitrarily additively constructible out of uniform random variables of appropriate width θ, it is $(\epsilon_2 = \frac{\epsilon'}{\log_2 A})$-approximately additively constructible by definition. Then, by Proposition 4.4.1, we know that we can choose θ so that regardless of the N we choose, our procedure satisfies Property 4.4.1. Furthermore, this particular choice for ϵ_2 also means that the last term the sum bounding the average variable rate is bounded above by ϵ'.

Next, we can set $\epsilon_3 = \epsilon'$ and calculate an N_3 such that if

$$N > N_3 = \frac{2\log_2\left(\frac{1}{\theta}\right) + \frac{\epsilon'}{\log_2 A}\log_2\left(\frac{1}{\epsilon'}\right)}{\epsilon'} \tag{4.14}$$

we have:

$$\frac{2\log_2\left(\frac{1}{\theta}\right) + \epsilon_2\log_2\left(\frac{1}{\epsilon_3}\right)}{N} < \epsilon'$$

Similarly, we can calculate N_1 for the first term and see that if

$$N > N_1 = \frac{\frac{1}{\ln 2} + 5 + \log_2(E\left[W_t^2\right]) + 2\log_2 D}{\epsilon'} \tag{4.15}$$

we have

$$\frac{\frac{1}{\ln 2} + 5 + \log_2(E\left[W_t^2\right]) + 2\log_2 D}{N} < \epsilon'$$

The case for the second term is even easier since $\log_2 N \leq \sqrt{N}$ and so:

$$N > N_2 = \sqrt{\frac{1}{\epsilon'}} \tag{4.16}$$

implies that

$$\frac{\log_2 N}{N} < \epsilon'$$

We can combine (4.15), (4.16), and (4.14) and know that as long as $N > \max(N_1, N_2, N_3)$, the average rate of the variable rate portion will not exceed $4\epsilon' = \frac{4}{6}\epsilon$. Furthermore, since $\epsilon_3 = \epsilon'$, we know that the average distortion on the $\hat{X}$ is going to be below $(D + \epsilon')$.

Since $R_\infty^X(D) = \liminf_{N\to\infty} R_N^X(D)$, we know that we can find $N > \max(N_1, N_2, N_3)$ so that $R_N^X(D) \leq R_\infty^X(D) + \epsilon'$. Property 4.4.1 tells us that for this N, the transformed $\{\vec{Y}_i\}$ look like i.i.d. samples of the finite horizon problem with horizon N. So, the classical rate-distortion theorems tell us that we can find an M large enough so that we can encode the superblock $\vec{Y}_{kM}^{kM+M-1}$ with rate less than $(R_N^X(D) + \epsilon') \leq (R_\infty^X(D) + 2\epsilon')$ per sample of X to a fidelity $D + \epsilon'$. For the distortion this means that since:

$$E\left[\frac{1}{M}\sum_{i=1}^{M} \|\vec{Y}_i - \hat{\vec{Y}}_i\|^2\right] \leq N(D + \epsilon')$$

we know that:

$$E[\left[\frac{1}{NM}\sum_{i=1}^{NM} (X_{kMN+i} - \hat{X}_{kMN+i})^2\right] \leq D + 2\epsilon'$$

for every $k > 0$.

Putting everything together, we then have average rate less than $(R_\infty^X(D) + 2\epsilon' + 4\epsilon') = (R_\infty^X(D) + \epsilon)$ and average distortion to within $D + 2\epsilon' < D + \epsilon$. The procedure has finite end-to-end delay by inspection and so the theorem is proved. □

4.5 From Variable Rate To Fixed Rate

The use of variable rate codes in Theorem 4.4.1 may be a bit troubling since we are mainly interested in fixed-rate codes which encode regularly sampled random processes into a regular stream of bits.

However, one can easily notice that since the blocks $\vec{Y}_{kM}^{kM+M-1}$ are independent of each other by construction (a simple consequence of Property 4.4.1) and our dithering process is designed to make the offsets look independent as well, that the sum of the length of the variable length encoded offsets is i.i.d. from superblock to superblock. This memorylessness at the superblock level, and the fact that the variable length segments are self-punctuating, suggests that this variable rate stream can be buffered (and possibly padded) in order to generate a fixed-rate stream. With sufficient delay added at the decoder, this fixed-rate stream can be used to recover the original variable rate stream well in time for successful decoding.

The only difficulty with this straightforward fixed-rate framing of a variable rate stream is in evaluating what happens in those rare cases where the fixed-rate stream lags too far behind the variable rate one. In such cases, we will not have enough information available at the decoder at the time when we are asked to output $\hat{X}_t$. Instead, the decoder will be forced to emit the best extrapolation based on what we have received so far. Although the resulting distortion may be very large, it is hoped that it will be compensated for by its rarity and so will not effect the average distortion by much.

4.5.1 Buffering and fixed-rate framing at the encoder

Let us be explicit about how the self-punctuated variable rate stream is buffered and transformed into a fixed-rate stream:

1. Incoming self-punctuating variable rate messages S_k of length L_k encoding the superblocks $\hat{X}_{kMN+1}^{kMN+MN}$ are buffered.

2. If the buffer length is less than MNR (presumed to be a positive integer for our convenience), then enough '0's are added (called 'padding') to the buffer to make it that length.

3. The oldest MNR bits are sent as part of the fixed-rate stream and are removed from the buffer. The process goes back to step 1 and repeats.

At the decoder, this buffering is undone assuming that the variable length stream was self-punctuating and that this punctuation was not disrupted by inserting a string of '0's between symbols. We also introduce a tolerable delay parameter T which tells how far in the past we want to reconstruct now.

1. Buffer incoming bits from the fixed-rate stream

2. If the $k-T$ message S_{k-T} is complete and available in the buffer, decode $\hat{X}_{(k-T)MN+1}^{(k-T)MN+MN}$ and continue from step 1

3. If the S_{k-T} message is not completely in the buffer already, use everything we have received so far to compute our best estimate for the relevant $\hat{X}_{(k-T)MN+1}^{(k-T)MN+MN}$ and continue from step 1

The choice of fixed rate R must be at least slightly higher than the average rate of the variable rate code in order to be able to absorb the effect of the rate variations in the original code. In fact, we would like the fixed rate code to have the following property:

Property 4.5.1 *The fixed rate code is such that:*

- *("Rare Overflows") The probability that the message S_{k-T} is not completely in the decoding buffer at arbitrary k goes to zero as T tends to infinity.*

- *("Light Tails") There exists some $\epsilon > 0$ so that for every H, the probability that the message $S_{k-(T+H+1)}$ is not completely in the decoding buffer at arbitrary k is less than $\left((A+\epsilon)^2\right)^{-MN}$ times the probability that the message $S_{k-(T+H)}$ is not completely in the decoding buffer at k.*

The first part of the property tells us that "overflows" can be made arbitrarily rare by increasing end-to-end delay. The second part of the property tells us that larger overflows are much rarer than smaller ones. This lets us guarantee that the occasional overflows do not end up dominating the expected squared error distortion in the sum as shown by the following lemma.

Lemma 4.5.1 *If the fixed rate R code corresponding to a variable rate code of Theorem 4.4.1 with expected average distortion D satisfies Property 4.5.1, then for all $\delta > 0$ we can choose a T so that the expected distortion is less than $D(1+\delta)$.*

Proof: For notational clarity, we will use $\tilde{X}$ to refer to the reconstructions from the fixed-rate code and $\hat{X}$ to refer to those from the original variable rate code: The expected distortion on a block is:

$$\begin{aligned}
& E\left[\frac{1}{N}\sum_{j=0}^{N-1}(X_{kMN+iN+j} - \tilde{X}_{kMN+iN+j})^2\right] \\
= \;& P(\text{no overflow})E\left[\frac{1}{N}\sum_{j=0}^{N-1}(X_{kMN+iN+j} - \tilde{X}_{kMN+iN+j})^2\,|\text{no overflow}\right] \\
& + \sum_{l=1}^{\infty} P(\text{overflow} = l)E\left[\frac{1}{N}\sum_{j=0}^{N-1}(X_{kMN+iN+j} - \tilde{X}_{kMN+iN+j})^2\,|\text{overflow} = l\right] \\
< \;& DP(\text{no overflow}) + \sum_{l=1}^{\infty} P(\text{overflow} = l)(D+K)A^{2MNl}
\end{aligned}$$

where we define

$$\begin{aligned}
K &= \sum_{l=0}^{\infty} A^{-2l}E\left[W^2\right] \\
&= \frac{E\left[W^2\right]}{1-A^{-2}}
\end{aligned}$$

to help bound the additional predicted squared error introduced by the driving noise that we have not accounted for at all in the extrapolations. Continuing, we get:

$$\begin{aligned}
& E\left[\frac{1}{N}\sum_{j=0}^{N-1}(X_{kMN+iN+j} - \tilde{X}_{kMN+iN+j})^2\right] \\
\leq \;& DP(\text{no overflow}) + (d+K)\sum_{l=0}^{\infty}\left((A+\epsilon)^{-2MN}\right)^l P(\text{overflow} = 1 \text{ superblocks})A^{2MNl} \\
\leq \;& DP(\text{no overflow}) + P(\text{overflow})(D+K)\sum_{l=0}^{\infty}\left((\frac{A}{A+\epsilon})^{2MN}\right)^l \\
\leq \;& DP(\text{no overflow}) + P(\text{overflow})\frac{D+K}{1-\left((\frac{A}{A+\epsilon})^{2MN}\right)} \\
< \;& D\left(1 + \frac{1+\frac{K}{D}}{1-\left((\frac{A}{A+\epsilon})^{2MN}\right)}P(\text{overflow})\right)
\end{aligned}$$

Property 4.5.1 assures us that we can choose a T large enough so that $P(\text{overflow}))$ is small enough to assure that $\frac{1+\frac{K}{D}}{1-((\frac{A}{A+\epsilon})^{2MN})}P(\text{overflow}) \leq \delta$. $\square$

The problem then reduces to seeing whether Property 4.5.1 can be achieved by a fixed rate code.

4.5.2 The difficulty with the original code: too much variation

We now examine the variable rate source code of Theorem 4.4.1. Recall from its proof and Proposition 4.4.3 that the variable rate portion has:

$$\begin{aligned}\frac{E[L_t]}{N} &< \left(\frac{\frac{1}{\ln 2} + 5 + \log_2(E\left[W_t^2\right]) + 2\log_2 D}{N}\right) \\ &\quad + \left(\frac{\log_2 N}{N}\right) \\ &\quad + \left(\frac{2\log_2\left(\frac{1}{\theta}\right) + \epsilon_2 \log_2\left(\frac{1}{\epsilon_3}\right)}{N}\right) \\ &\quad + \epsilon_2 \log_2 A\end{aligned}$$

In this expectation, the $\frac{1}{\ln 2}$ comes from the exponentially distributed $\tilde{L}_t$ and the terms multiplied by ϵ_2 represent the contribution to the expected length coming from the constant-sized random variable (a multiple of B_t) which is present with probability ϵ_2. The $\tilde{L}_t$ are independent from superblock to superblock, while the B_t are all independent.

The $\epsilon_2 \log_2 A$ portion coming from the B_t term is of particular concern to us since it is of the same order as the fixed-rate R itself while the other terms all die away with larger N. For the moment, concentrate only on this part. Suppose that the rate we allocate in the fixed rate code is $C\epsilon_2 \log_2 A$ with $C > 0$ and $C\epsilon_2$ still small relative to 1. Since the variable rate part occurs only every N time steps, this gives us a total rate of $CN\epsilon_2 \log_2 A$ for each sample which takes 0 bits with probability $(1 - \epsilon_2)$ and takes $N \log_2 A$ with probability ϵ_2. Using Chernoff's bound, we can calculate the probability that the sum of j independent such random variables will overflow by:

$$\begin{aligned}P(\text{overflow}) &= P(\sum_{t=1}^{j} N \log_2 AB_t \geq jCN\epsilon_2 \log_2 A) \\ &= P(\sum_{t=1}^{j} B_t \geq jC\epsilon_2) \\ &\leq \inf_{s\geq 0}\left((1-\epsilon_2) + \epsilon_2 e^s\right)^j e^{-sjC\epsilon_2} \\ &= \left(\inf_{s\geq 0}(1-\epsilon_2)e^{-sC\epsilon_2} + \epsilon_2 e^{-s(C\epsilon_2 - 1)}\right)^j\end{aligned}$$

We can calculate the minimum by taking the derivative with respect to s and setting it to zero giving us:

$$-C\epsilon_2(1-\epsilon_2)e^{-sC\epsilon_2} + -(C\epsilon_2 - 1)\epsilon_2 e^{-s(C\epsilon_2-1)} = 0$$

which has a unique solution:

$$s = \ln\left(\frac{C(1-\epsilon_2)}{1 - C\epsilon_2}\right)$$

which also minimizes the expression and results in an overflow probability κ^j that tends to 0 exponentially with increasing j.

However, our interest is also in the relative probabilities of different sizes of the overflows. Focusing only on the $(N \log_2 A)B_t$ term, we notice that in order for the sum of j such variables to overflow more than H superblocks, the sum must overflow by more than $HMNR$. Putting it together and using the argument above, we get:

$$\begin{aligned}
& P(\text{overflow more than } H \text{ superblocks}) \\
= \; & P(\sum_{t=1}^{j} N \log_2 AB_t \geq jCN\epsilon_2 \log_2 A + HMNR) \\
= \; & P(\sum_{t=1}^{j} B_t \geq jC\epsilon_2 + HM\frac{R}{\log_2 A}) \\
\leq \; & \inf_{s \geq 0} ((1-\epsilon_2) + \epsilon_2 e^s)^j \, e^{-s(jC\epsilon_2 + HM\frac{R}{\log_2 A})} \\
= \; & \inf_{s \geq 0} \left(((1-\epsilon_2) + \epsilon_2 e^s)e^{-sC\epsilon_2}\right)^j \, e^{-sH\frac{MR}{\log_2 A}}
\end{aligned}$$

We can plug in the s value from the previous minimization and thereby get a slightly loose bound[1]:

$$\begin{aligned}
& P(\text{overflow more than } H \text{ superblocks}) \\
\leq \; & \kappa^j e^{-H\frac{MR}{\log_2 A}} \\
= \; & \kappa^j \left(\frac{1-C\epsilon_2}{C(1-\epsilon_2)}\right)^{H\frac{MR}{\log_2 A}} \\
= \; & \kappa^j \left(\left(\frac{1}{C(1-\epsilon_2)} - \frac{\epsilon_2}{1-\epsilon_2}\right)^{\frac{MR}{\log_2 A}}\right)^H
\end{aligned}$$

This also clearly drops exponentially in the amount of the overflow, but the question is whether it drops fast enough to satisfy the "Light Tails" part of Property 4.5.1. For that, we need to be able to get some $\epsilon > 0$ so that:

$$\left(\frac{1}{C(1-\epsilon_2)} - \frac{\epsilon_2}{1-\epsilon_2}\right)^{\frac{MR}{\log_2 A}} \leq \left((A+\epsilon)^2\right)^{-MN}$$

The M in the exponents cancel out, but even then, we are left with:

$$\left(\frac{1}{C(1-\epsilon_2)} - \frac{\epsilon_2}{1-\epsilon_2}\right) \leq \left((A+\epsilon)^{2\frac{\log_2 A}{R}}\right)^{-N}$$

Since we expect both $C\epsilon_2$ and ϵ_2 to be small, this means that we roughly need C to be of the same order as $\left((A+\epsilon)^{2\frac{\log_2 A}{R}}\right)^{N}$ which means that ϵ_2 must be much smaller than $\left((A+\epsilon)^{2\frac{\log_2 A}{R}}\right)^{-N}$. In order to make ϵ_2 that small, θ needs to be small too. But even if we assume that the density for W_t is Lipshitz, Theorem C.2.2 only lets us establish a linear

[1] See Section 7.2.1 for a tight one since this is effectively an embedded erasure channel with feedback. Using the tight bound does not change the essential flaw of this code.

relationship between θ and ϵ_2. So, an exponentially small ϵ_2 would necessitate a substantial $\log_2 \frac{1}{\theta}$ and prevent us from approaching the rate-distortion limits in expected rate.

Although the value for s we used was conservative, it turns out that even the optimal value for s is only different by a small constant factor. This problem is fundamental. The variation in rate introduced by the B_t terms is too much to be safely tamed by a fixed-rate code.

4.5.3 A "less variable" variable rate code

Fortunately, there is a way of adjusting our variable rate code and its analysis to eliminate the variation caused by the B_t. Recall that the purpose of the B_t is to allow us to perfectly simulate the W_t. In Figure 4-3, this allowed us to use a vector quantizer tailored to superblocks consisting of M independent samples of the finite N horizon version of the original source.

Without the B_t, the superblocks would still consist of the M independent samples, but they would be samples of something which had only approximately the same distribution as the finite N horizon version of the original source. Lemma C.2.4 lets us establish that we have the following relation between the rate-distortion performance of the two:

Lemma 4.5.2 *Let $\{X_t\}$ be the original finite N horizon source driven by $\{W_t\}$ and let $\{X_t'\}$ be the same, but driven by $\{W_t'\}$ where $W_t' = W_t$ except for W_0'. Let $W_1 = (1-B_\epsilon)W_1' + B_\epsilon Z$ in the sense of distribution where B_ϵ is an independent Bernoulli random variable. Then:*

$$R_N^{X'}(\frac{d}{1-\epsilon}) \le \frac{1}{1-\epsilon} R_N^X(d)$$

Proof: Let $\vec{Y}$ be the vector representing X_1^N and $\vec{Y}'$ be the vector representing ${X'}_1^N$. Then, by Lemma C.2.3, we know that we can define an appropriate $\vec{Z}$ so that

$$\vec{Y} = (1-B_\epsilon)\vec{Y}' + B_\epsilon \vec{Z}$$

Now, recall that:

$$R_N^{X'}(\frac{D}{1-\epsilon}) = \frac{1}{N} \inf_{p_{\hat{\vec{Y}}|\vec{Y}'} : E\left[\rho(\vec{Y}', \hat{\vec{Y}})\right] \le \frac{ND}{1-\epsilon}} I(\vec{Y}'; \hat{\vec{Y}}) \tag{4.17}$$

and:

$$R_N^X(D) = \frac{1}{N} \inf_{p_{\hat{\vec{Y}}|\vec{Y}} : E\left[\rho(\vec{Y}, \hat{\vec{Y}})\right] \le ND} I(\vec{Y}; \hat{\vec{Y}}) \tag{4.18}$$

Let p be a transition measure which satisfies the expected distortion conditions and gets within some δ of the infimum in (4.18). We can use this transition rule for the $\vec{Y}'$ and Lemma C.2.4 tells us that:

$$\begin{aligned} E\left[\rho(\vec{Y}', \hat{\vec{Y}})\right] &\le \frac{1}{1-\epsilon} E\left[\rho(\vec{Y}, \hat{\vec{Y}})\right] \\ &= \frac{ND}{1-\epsilon} \end{aligned}$$

and so it is a valid transition rule for (4.17). It need not be the infimizer and so we get:

$$\begin{aligned}
R_N^{X'}(\frac{D}{1-\epsilon}) &\leq \frac{1}{N}I(\vec{Y}';\hat{\vec{Y}}) \\
&\leq \frac{1}{1-\epsilon}\frac{1}{N}I(\vec{Y};\hat{\vec{Y}}) \\
&\leq \frac{1}{1-\epsilon}(R_N^X(D)+\delta)
\end{aligned}$$

Since δ was arbitrary, the lemma is proved. □

This means that we can choose ϵ_2 small enough so that the fixed rate part of the variable rate code of Figure 4-3 is within $\frac{1}{1-\epsilon_2}$ of the average rate required for the code with the B_t terms in it. A quick application of Proposition 4.4.2 gives us the following:

Proposition 4.5.1 *Consider W_t with finite variance and approximately additively constructible out of uniform random variables of width θ with ϵ_2 probability left over. Then, the fixed rate part of the code in Figure 4-3 has average rate which can be made to be arbitrarily close to $\frac{1}{1-\epsilon_2}R_N^X(D)$ as long as we are willing to accept average distortion $\frac{D}{1-\epsilon_2}$ per letter.*

Moreover, the length L_t of the variable rate encoding for $\check{X}_t+\xi_t$ under our new variable length encoding procedure has :

$$E[L_t] \leq \frac{1}{\ln 2}+5+2\log_2\left(\frac{1}{\theta}\right)+\log_2 N+2\log_2\frac{D}{1-\epsilon_2}+\log_2(E\left[W_t^2\right])$$

and furthermore, the probability for large deviations drops as follows:

$$P(L_t \geq \lambda) \leq P(\bar{L}_t \geq \lambda)$$

where:

$$\bar{L}_t = 7+2\log_2\left(\frac{1}{\theta}\right)+\log_2 N+2\log_2 D+\log_2(E\left[W_t^2\right])+\tilde{L}_t$$

and $\tilde{L}_t$ has exponential distribution: $P(\tilde{L}_t \geq \lambda) = 2^{-\lambda}$

Proof: Everything follows directly from the arguments above and Proposition 4.4.2. The only distinction is that we use $\frac{D}{1-\epsilon_2}$ in place of D in some places and bound the logarithm of this by $1+\log_2 D$ since ϵ_2 can be assumed to be small. □

4.5.4 Dealing with the remaining variability

The only question now is whether the sums of the $\tilde{L}_t$ are such that they satisfy Property 4.5.1. We know that the $\tilde{L}_t$ from different superblocks are independent from each other by construction. Within a superblock, the situation can get trickier, but let us ignore that for a moment and just assume that everything is independent.

Assume that we allocate an extra $\epsilon_4 N$ bits for each $\tilde{L}_t$. Then:

$$\begin{aligned}
P(\text{overflow}) &= P(\sum_{t=1}^{j}\tilde{L}_t \geq jN\epsilon_4) \\
&\leq \inf_{s\geq 0}\left(\frac{1}{(\ln 2-s)\ln 2}e^{-sN\epsilon_4}\right)^j
\end{aligned}$$

Taking derivatives and minimizing gives us a unique solution:

$$s = \ln 2 - \frac{1}{N\epsilon_4} \tag{4.19}$$

which we can plug back in to get:

$$P(\text{overflow}) \leq \left(\frac{e}{\ln 2}(N\epsilon_4)2^{-N\epsilon_4}\right)^j$$

This clearly tends to 0 with increasing j as long as $N\epsilon_4$ is large enough.

However, our interest is also in the relative probabilities of different sizes of the overflows. As before, we get:

$$\begin{aligned} & P(\text{overflow more than } H \text{ superblocks}) \\ = \; & P(\sum_{t=1}^{j} \tilde{L}_t \geq jN\epsilon_4 + HMNR) \\ \leq \; & \inf_{s \geq 0} \left(\frac{1}{(\ln 2 - s)\ln 2} e^{-sN\epsilon_4}\right)^j e^{-sHMNR} \end{aligned}$$

We can plug in the s value from the previous minimization and get:

$$\begin{aligned} & P(\text{overflow more than } H \text{ superblocks}) \\ \leq \; & \left(\frac{e}{\ln 2}(N\epsilon_4)2^{-N\epsilon_4}\right)^j e^{-(\ln 2 - \frac{1}{N\epsilon_4})HMNR} \\ = \; & \left(\frac{e}{\ln 2}(N\epsilon_4)2^{-N\epsilon_4}\right)^j \left((e^{R\ln 2 - \frac{R}{N\epsilon_4}})^{-MN}\right)^H \end{aligned}$$

This also drops exponentially in the amount of the overflow. However, to satisfy Property 4.5.1, we must have some ϵ so that:

$$e^{R\ln 2 - \frac{R}{N\epsilon_4}} \geq (A + \epsilon)^2$$

We can choose $N\epsilon_4$ large enough and ϵ small enough to do this whenever we have:

$$R > 2\log_2 A \tag{4.20}$$

Given that $R \geq \log_2 A$ is a necessary condition for finite distortion, condition (4.20) is not that demanding. In fact, this condition (4.20) can be tightened up by using a less wasteful universal code for the positive integers. Above, we use $2\log_2 \zeta$ bits to store an integer of size ζ. This can be brought down to $\log_2^* \zeta$ where $\log_2^*$ represents the sum of iterated logarithms.[11] $\log_2^*$ is less than any constant $K > 1$ times the logarithm and therefore results in a $\tilde{L}_t$ with distribution having $P(\tilde{L}_t \geq l) \leq \left(2^{\frac{2}{K}}\right)^{-l}$ even if all we had was a Chebychev inequality bound for the probability of large distortions. This would serve to move the $\ln 2$ over to be closer to $2\ln 2$ in equation (4.19) for the minimizing s and thereby reduce the condition (4.20) to give:

$$R > \log_2 A$$

This is tight since all codes must use least that much rate.

The only issue which remains is the question of the effective independence of the $\tilde{L}_t$ within a superblock.

4.6 Discussion

In this chapter, we have given a simple causal code that achieves the fundamental sensitivity bound for unstable scalar processes driven by bounded noise. We have also resolved a long-standing open problem by giving a variable rate code which approaches the rate-distortion performance limit for the infinite-horizon average squared-error distortion problem on exponentially unstable scalar Markov processes. Furthermore, we suggest that our initial variable rate code is fundamentally "too variable" to be reinterpreted as a fixed rate code and have argued that other codes exist which are far less variable. Even so, many open questions remain and we intend to resolve them in the future.

4.6.1 More General Sources and Distortions

It should be clear that the proof of Theorem 4.4.1 does not rely crucially on the fact that $\{X_t\}$ is scalar or that we are using the squared error distortion. The core of the argument is that the distortion measure penalizes larger differences more than smaller ones in a polynomial way. Via the logarithms involved in universally encoding integers, this gives us a nice exponential distribution for the variable length part of the code. It therefore seems clear that everything still works in the η-distortion case as long as $\eta > 0$. In vector cases, as long as the distortion measure has "full rank" in some sense, things should continue to work.

If X was a finite l-dimensional vector, then we could apply a vector version of the θ-dithering to each of its components and do the reconstructions in an analogous way. The only difference would be a factor of l increase in the average contribution of the variable rate component of the code. But since this component of the overall rate can be made arbitrarily small, this constant factor increase does not change the flavor of the result. In the same way, if the process $\{X_t\}$ was not Markov and was instead auto-regressive to some finite order, then we could apply the same sort of argument on a state-augmented version of X which was Markov. So, the theorem, though strictly proved above for only the scalar Markov case, holds more generally of vector valued finitely autoregressive processes.

Furthermore, we suspect that a minimal canonical state-space realization argument would extend this to any signal model within the autoregressive moving average (ARMA) class, as long as it can be finitely parametrized. Infinite order unstable models are likely going to be much more problematic. Time-varying linear and nonlinear models might also be somewhat difficult without a good way of bounding the extent of the time variation.

Of course, we would also like to see these ideas applied to real world sources rather than just theoretical models. We suspect that the arguments given here can be used in video-coding to remove the need for regular synchronization frames as a way to prevent coding artifacts from accumulating. This might give a major savings in the bitrate required for slowly varying scenes. In addition to multimedia signals, it will be interesting to apply these ideas to the encoding of other nonstationary stochastic signals where delays are a matter of concern.

4.6.2 Sensitivity of Near Optimal Codes

A natural question is how sensitive the codes in the above sections are to bit errors in the past. Theorem 3.3.2 already tells us that $\Delta^+(d)$ must be growing exponentially at least as fast as $A^{2d} = 2^{2\log_2 A}$ when viewed under the squared-error distortion measure. The only question is whether it is very much faster than that.

Since the $\check{X}_t$ are used to make the sub-blocks of size N look independent of each other in Figure 4-4, we know that any long range propagation of errors will happen through the $\check{X}_t$ stream which is recursively encoded unlike the $\hat{\vec{Y}}$. The tricky part is the lossless code for the integers. That code encodes possibly exponentially large quantities in a linear number of bits and is hence very sensitive to bit errors. If we are not careful, the sensitivity of the whole code could be very large.

It seems likely that if we assume that the W_t have bounded support, then the decoder can detect when an error is placing us completely out of bounds and can thereby limit this exponential problem. With those limits in place, the code should have a $\Delta^+(d)$ which is close to the fundamental bound given by Theorem 3.3.2, only with much larger constants.

In general, removing the bounded noise assumption should be a goal and we suspect that it is possible with proper assumptions on the tails of the driving noise distributions. After all, Nair [45] is able to track a system driven by unbounded noise in the estimation setting and we suspect those arguments would work here as well.

4.6.3 The Fixed Rate Case

We strongly suspect that there is a way around the currently unjustified assumption that the errors on the encodings of sub-blocks within the same superblock are independent. A naive bound will not suffice since we currently have no limit on how large M can be as a function of N and D. If the size of M can be kept small, even the simple bounds will work. This will probably require a closer look at the properties of the vector quantizer we are using on the superblocks. If M cannot be made small, we might be able to limit the probability that the reconstructions are very bad on many sub-blocks at once. As long as we can avoid such cases, we should be fine.

Chapter 5

Reliable Communication of "Weak Bits"

We have seen that the main problem we encounter in attempting to track an unstable process across a noisy channel is the steady accumulation of errors. This accumulation is unacceptable for bit-streams resulting from weak source codes. To allow ourselves to reliably transport weak bit-streams, we have to generalize our concept of a decoder. Having done that, we introduce a new stronger sense of reliable transmission and an operational notion of capacity, which we call anytime capacity, that goes with it.

5.1 Anytime Decoding

Definition 5.1.1 *Let τ be the sampling time of the channel and θ the relative offset of the bit-stream that entered the encoder. For a positive real number R, a* rate R bits per unit time anytime channel decoder *is a sequence $\mathcal{D}^a$ of functions $\mathcal{D}^a_i$ from $\mathcal{B}^i$ into $\{0,1\}^{\lfloor(\theta+i)R\tau\rfloor}$.*

Rather than producing a single output stream, these anytime channel decoders give updated estimates for all previous input bits with each received symbol from the channel. This eliminates the question of which output bit corresponds to which input bit and in principle it allows the system at the output of the channel decoder to decide (possibly adaptively) how much it is willing to wait for a given bit.

Definition 5.1.2 *An* output bit delay selector *is a sequence d_1^∞ of non-negative real numbers that combines with an anytime decoder to generate an output bit sequence as follows: $\hat{s}_i = (\mathcal{D}^a_j(b_1^j))_i$ where $j = \left\lceil \frac{(i-\theta)}{R\tau} + \frac{d_i}{\tau} \right\rceil$. If $d_i = d$ for all i then we call it a* fixed delay selector of delay *d and use the shorthand notation $d_1^\infty = d$.*

The d_i is used to select the delay (in units of time) experienced before reconstructing bit i. The definition that we have used for delay might seem a little odd but it is motivated by the idea that zero delay should correspond to the earliest possible time that the decoder could have received any signal causally dependent on the bit in question.

With a specification of delay, we can evaluate the performance in terms of probability of error at any time naturally as follows:

$$P_{\text{error}}(\mathcal{E}, \mathcal{D}^a, d_i, i) = E_{\left\{S_1^{\left\lfloor \left\lceil \frac{(i-\theta)}{R\tau} + \frac{d_i}{\tau} \right\rceil R\tau \right\rfloor}\right\}} \left[P(\hat{S}_i \neq S_i) \right]$$

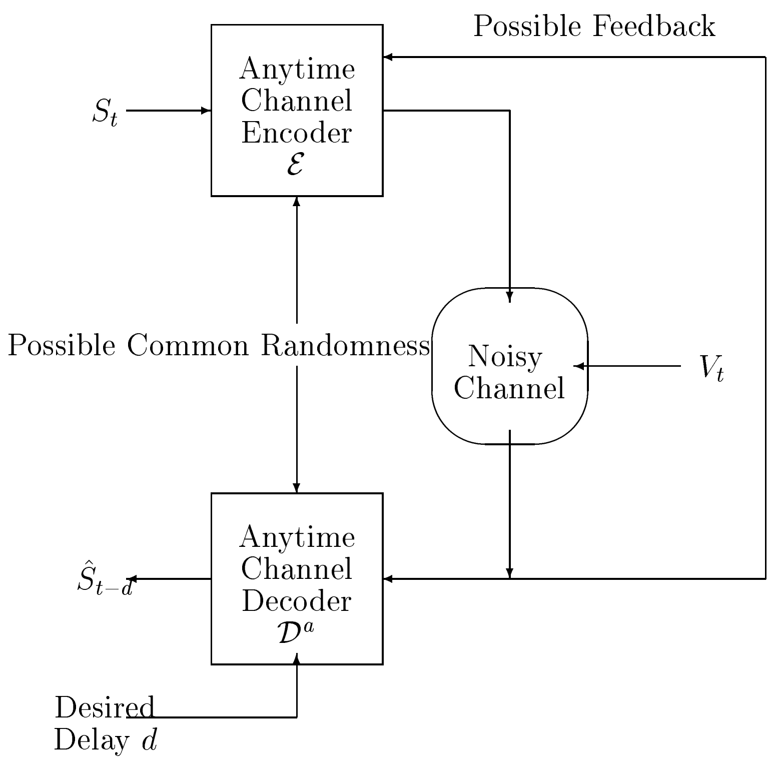

Figure 5-1: Anytime encoders and decoders

and can also consider the probability of error independent of time:

$$P_{\text{error}}(\mathcal{E}, \mathcal{D}^a, d_1^\infty) = \sup_{i \geq 0} P_{\text{error}}(\mathcal{E}, \mathcal{D}^a, d_i, i)$$

Here, the probability measure used for the expectation is over the binary strings S_1^N and is assumed to be uniform (i.e. each bit is equally likely to be a zero or a one and they are all independent of each other). The probability of erroneous decoding is calculated based on all the remaining randomness in the system. If the encoders and decoders are deterministic, this means the randomness is introduced by the channel. If we allow common randomness at the encoder and decoder (by means of a common dependence on an additional random variable), that is used in computing the probability of error as well.

To see that anytime decoders and delay selectors are truly a generalization of our other definition of a channel decoder, consider the following proposition.

Proposition 5.1.1 *For any θ offset, rate R channel decoder $\mathcal{D}$ with reconstruction profile r_1^∞, there exists a rate R anytime channel decoder $\mathcal{D}^a$ and output bit delay selector d_1^∞ that generates the same set of output bits at the same times.*

Proof: We just construct the anytime decoder and delay selector as follows:

$$\begin{aligned}
\left(\mathcal{D}_j^a(b_1^j)\right)_i &= \begin{cases} \mathcal{D}_{r_i}(b_1^{\lfloor \frac{(\theta+r_i)}{R\tau} \rfloor}) & \text{if } \lfloor \frac{(\theta+r_i)}{R\tau} \rfloor \leq j \\ 0 & \text{otherwise} \end{cases} \\
d_i &= \frac{2\theta + r_i - i}{R}
\end{aligned}$$

It is easy to see that this construction gives the desired result. □

5.2 Anytime Capacity

Proposition 5.1.1 above does not give us anything new since the resulting anytime decoder never repairs any errors made at previous time steps. To explicitly consider that possibility, we introduce a new decoder:

Definition 5.2.1 *The* maximum likelihood anytime decoder $\mathcal{D}^a$ based on encoder $\mathcal{E}$ *is defined as follows:*

$$\mathcal{D}_i^a(b_1^i) = \arg \max_{s_1^{\lfloor (\theta+i)R\tau \rfloor} \in \{0,1\}^{\lfloor (\theta+i)R\tau \rfloor}} P(B_1^i = b_1^i \,\middle|\, A_1^i = \mathcal{E}(s_1^{\lfloor (\theta+i)R\tau \rfloor}))$$

Estimating bits on the basis of maximum likelihood (completely in the spirit of sequential decoding [71]) is clearly the best that any decoder could do given no further information about S_1^∞. This allows initially incorrect estimates to subsequently be corrected and lets us to define a new notion of reliable transmission that explicitly considers the rate at which past errors get corrected.

Let $f(d) > 0$ be any decreasing function of delay d. We say that an anytime encoder and decoder pair $(\mathcal{E}, \mathcal{D}^a)$ achieves $\mathcal{R}_{\text{anytime}}(f)$ iff there exists a finite constant $K > 0$ so that the probability of error for every bit position decays with delay at least as fast as $Kf(d)$. This notion of reliable transmission naturally induces a kind of capacity:

Definition 5.2.2 *The* f-anytime capacity $C_{anytime}(f)$ *of a channel is the least upper bound of the rates at which the channel can be used to transmit data with an arbitrarily small probability of error that decays with delay at least as fast as the function* f *does.*

$$C_{anytime}(f) = \sup\{R|\exists(K > 0, Rate(\mathcal{E}, \mathcal{D}^a) = R)\, \forall d > 0\; P_{error}(\mathcal{E}, \mathcal{D}^a, d_1^\infty = d) \leq Kf(d)\}$$

The idea is that f is a function of delay d which decays to zero as $d \to \infty$. We want more reliability for older bits than for the more recent ones. While this may superficially remind us of the prior work on unequal error-protection codes[47], it is quite different since in our formulation every bit in the stream starts out recent but eventually becomes old. In this sense, all bits are fundamentally equal even though at any given time, they are treated differently. Our work here is closest in motivation to the work of Krich [40, 41, 42] that we mentioned earlier.

For most of this chapter, we use an exponential rate of decay because it is convenient for our purposes. Instead of parametrizing the anytime capacity with a general function $f(d)$, we use a scalar α which tells us how fast the exponential decays with d.

Definition 5.2.3 *The* α-anytime capacity $C_{anytime}(\alpha)$ *of a channel is the least upper bound of the rates at which the channel can be used to transmit data with an arbitrarily small probability of error that decays with delay at least exponentially at a rate* α*.*

$$C_{anytime}(\alpha) = C_{anytime}(2^{-\alpha d})$$

The definition parametrized by the scalar α should be reminiscent of definition 2.2.4 for the reliability function. In our definition of anytime capacity, however, we hold the encoder and anytime decoder constant depending on the rate R, while allowing ourselves to adjust the probability of error purely by the choice of delay selector. Moreover, we want our probability of error to be less than $K2^{-\alpha d}$ *for every delay* d and *for every bit* rather than just for a single d or a subset of bits. Thus, our encoders need to be "delay universal" in this sense.

We also do not restrict ourselves to any particular structure (like block or convolutional) on the encoder *a priori*. Finally, this definition of anytime capacity is an operational one. It tells what we mean but does not give us a way of calculating its value. The interesting aspect of the definition is the new and stronger sense of "reliable transmission" that the function f introduces. The shift is conceptual. Rather than viewing the rate at which the probability of error goes to zero as a proxy for complexity or as a way of evaluating a family of codes, we are demanding that the probability of error goes to zero for every bit in a given code! This is certainly related to the work on the bounded-time decoding of convolutional codes in [19, 20]. The difference is that we require the probability of error to go to zero as delay tends to infinity while in the case of bounded-time decoding, it tended to the probability of error of the underlying finite constraint-length convolutional code.

An interesting property which illustrates the power of our new sense of reliable transmission is the following:

Lemma 5.2.1 *If* $\sum_d f(d) < \infty$*, almost every bit is eventually decoded correctly. Let* $\tilde{S}_j^d = \hat{S}_j$ *when using the fixed delay selector* d*. For any* $f(d)$ *that is summable, if* $R < C_{anytime}(f)$ *then* $\exists(\mathcal{E}, \mathcal{D}^a)$ *such that for all* $j > 0$ *the sequence* $[\tilde{S}_j^0, \tilde{S}_j^1, \ldots]$ *eventually converges to the correct value* S_j *and stays there with probability one.*

Proof: We will use the encoder and anytime decoder from the definition of f-anytime capacity. To show that it is eventually correct, we only need to show that with probability one, it is wrong only a finite number of times. Let W_j be the random variable that represents the number of entries in the sequence $(\tilde{S}_j^0, \tilde{S}_j^1, \ldots)$ that are wrong. So $W_j = \sum_{d=0}^{\infty}(1 - \delta(\tilde{S}_j^d, S_j))$ where δ is the usual Kronecker delta function. Then, taking expectations:

$$\begin{aligned}
E[W_j] &= E\left[\sum_{d=0}^{\infty}(1 - \delta(\tilde{S}_j^d, S_j))\right] \\
&= \sum_{d=0}^{\infty} E\left[(1 - \delta(\tilde{S}_j^d, S_j))\right] \\
&= \sum_{d=0}^{\infty} E\left[P(\tilde{S}_j^d \neq S_j)\right] \\
&\leq \sum_{d=0}^{\infty} P_{\text{error}}(\mathcal{E}, \mathcal{D}^a, d_1^{\infty} = d) \\
&< \sum_{d=0}^{\infty} K f(d) \\
&< \infty
\end{aligned}$$

Since W_j has a finite expectation and is a positive random variable by construction, it has a zero probability of being infinite. Hence, with probability one, for any j our estimates will be wrong only a finite number of times $[T_1, T_2, \ldots, T_{N(\omega)}](\omega)$. This finite set always has a largest element $T_N(\omega)(\omega)$, and hence all $\tilde{S}_j^d(\omega) = S_j$ as long as $d > N(\omega)$. □

It should be clear that Lemma 5.2.1 applies in all the exponential cases of anytime capacity as long as $\alpha > 0$. With this, we can relate our new notion of reliable transmission to the traditional ones. Clearly, $C_0 \leq C_{\text{anytime}}(f) \leq C$ for all $f(d)$ which have $\lim_{d \to \infty} f(d) = 0$. The main difference between zero-error capacity and anytime capacity is that zero-error capacity requires us to know in advance exactly when the bit is going to be received correctly. Anytime capacity is not as stringent and lets the time of final correct arrival be somewhat random and to vary based on what the channel actually does (and possibly the input sequence itself). In particular, we can easily see how for a binary erasure channel with noiseless feedback our simple buffering encoder described in Figure 2-1 and given by (2.2) and (2.3) can be used to get non-zero anytime capacity at an exponential rate of decay.

This sort of relaxation of Shannon's notion of zero-error capacity is also often justified in real applications. For example, it is standard for "error-free" reliable transmission protocols on the Internet to not guarantee a time of arrival for the transmitted bits.[51] In a sense, our definition of anytime capacity is the natural generalization of the way in which TCP/IP gives us reliable transmission.

5.3 Lower-Bounding Anytime Capacity without Feedback

Having a definition of anytime capacity is only interesting if it is non-zero in some interesting cases. The erasure case with noiseless feedback seems a little special. But in this section, we will show how it is possible to have a non-zero anytime capacity even for channels without any feedback at all! We will use a random coding argument using our definition

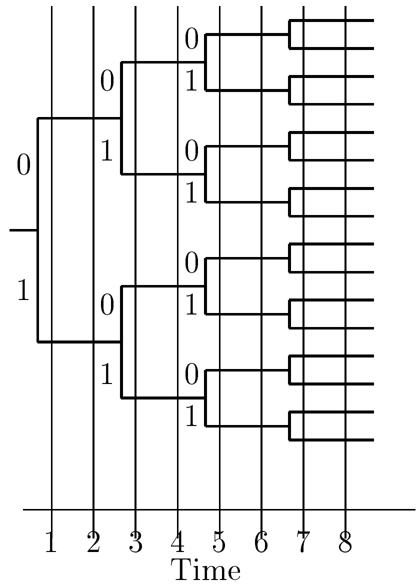

Figure 5-2: A general channel encoder viewed as a tree.

of encoders.[1] After illustrating how they work for the binary erasure and AWGN channel, we will use random encoders to prove Theorem 5.3.3 in which we show that for memoryless channels, the block random coding error exponent can be used to give a lower bound to the α-anytime capacity.

Definition 5.3.1 *A random rate* R offset θ *encoder* $\mathcal{E}$ drawn with probability $P(a)$ *is an encoder for which every possible channel input* $a_i = \mathcal{E}_i(s_1^{\lfloor(\theta+i)R\tau\rfloor})$ *is drawn from the channel input alphabet* $\mathcal{A}$ *according to the distribution* $P(a)$*. Furthermore,* $\mathcal{E}_i(x_1^{\lfloor(\theta+i)R\tau\rfloor})$ *is independent of* $\mathcal{E}_j(y_1^{\lfloor(\theta+j)R\tau\rfloor})$ *whenever either* $i \neq j$ *or* $x_1^{\lfloor(\theta+i)R\tau\rfloor} \neq y_1^{\lfloor(\theta+j)R\tau\rfloor}$.

As Figure 5-2 illustrates, this can be thought of as a labeled tree with an independent random variable drawn according to $P(a)$ sitting at every intersection of the tree with vertical lines drawn at $[\tau, 2\tau, \ldots]$. The input bits s_1^∞ are used to select a path through this tree, and the encoder emits the realizations of the random variables it sees along the path.

Our analysis of random encoders will tell us the expected behavior. There are two ways of interpreting this: either as the behavior of an encoder/decoder system with access to common randomness (ie. both the encoder and decoder are dependent on a common random input V), or as the expectation taken relative to a particular probability measure over the set of deterministic encoders.

5.3.1 Binary Erasure Channel

We will start with the binary erasure channel because it gives insight into how a random encoder combined with a maximum likelihood anytime decoder can actually work. The first

[1]Here one sees the connection of our encoders with the tree encoders of [71] and more recently in the Computer Science community: [58] and [52]; though they did not consider them as being actually infinite as we do. However, these are not related to Berger's tree codes in [6] which are not causally encoded.

thing we notice is the following.

Proposition 5.3.1 *For a binary erasure channel, and any* $b_1^i \in \{0,1,\emptyset\}^i$,

$$P(B_1^i = b_1^i \,\Big|\, A_1^i = a_1^i\,) = \begin{cases} 0 & \text{if } \exists j \leq i \; a_j \neq b_j \neq \emptyset \\ e^k(1-e)^{i-k} & \text{otherwise} \end{cases}$$

where k *is the number of erasures in* b_1^i.

Proof: This is a simple application of the definition of a memoryless binary erasure channel. The only errors possible are erasures and the probability of an observed sequence only depends on the number of erasures. □

This means that given an observed sequence b_1^i, all we can really do is rule out certain input sequences.

Definition 5.3.2 *If* $P(B_1^i = b_1^i \,\Big|\, A_1^i = \mathcal{E}(s_1^{\lfloor(\theta+i)R\tau\rfloor})\,) = 0$, *we call the sequence* $s_1^{\lfloor(\theta+i)R\tau\rfloor}$ incompatible with b_1^i. *If all possible extensions* $s_1^{\lfloor(\theta+i)R\tau\rfloor}$ *of a subsequence* s_1^j *(*$j < \lfloor(\theta+i)R\tau\rfloor$*) are incompatible with* b_1^i, *we call the subsequence* s_1^j incompatible with b_1^i *as well.*

The nice thing is that once a certain input sequence is ruled out, so are all extensions of it.

Proposition 5.3.2 *For all encoders E, if* s_1^j *is incompatible with* b_1^i, *it is also incompatible with all extensions* b_1^{i+l}.

Proof: This is a consequence of the binary erasure channel's memorylessness.

$$\begin{aligned} & P\left(B_1^{i+l} = b_1^{i+l} \,\Big|\, A_1^{i+l} = \mathcal{E}(s_1^{\lfloor(\theta+i+l)R\tau\rfloor})\right) \\ = \;& P\left(B_1^{i} = b_1^{i} \,\Big|\, A_1^{i+l} = \mathcal{E}(s_1^{\lfloor(\theta+i+l)R\tau\rfloor})\right) P\left(B_{i+1}^{i+l} = b_{i+1}^{i+l} \,\Big|\, A_1^{i+l} = \mathcal{E}(s_1^{\lfloor(\theta+i+l)R\tau\rfloor})\right) \\ = \;& P\left(B_1^{i} = b_1^{i} \,\Big|\, A_1^{i} = \mathcal{E}(s_1^{\lfloor(\theta+i)R\tau\rfloor})\right) P\left(B_{i+1}^{i+l} = b_{i+1}^{i+l} \,\Big|\, A_1^{i+l} = \mathcal{E}(s_1^{\lfloor(\theta+i+l)R\tau\rfloor})\right) \\ = \;& 0 \end{aligned}$$

Thus it is incompatible with the extension as well. □

We can now construct the following nested sets. Let $\mathcal{C}_j^i(b_1^i)$ be the set of possible $s_1^j \in \{0,1\}^j$ that are not incompatible with b_1^i. By construction, every element $s_1^{j+k} \in \mathcal{C}_{j+k}^i(b_1^i)$ is an extension of some $s_1^j \in \mathcal{C}_j^i(b_1^i)$. Conversely, every truncation s_1^j of $s_1^{j+k} \in \mathcal{C}_{j+k}^i(b_1^i)$ is an element of $\mathcal{C}_j^i(b_1^i)$. It is also clear from Proposition 5.3.2 that $\mathcal{C}_j^{i+l}(b_1^{i+l}) \subseteq \mathcal{C}_j^i(b_1^i)$. Furthermore, $\mathcal{C}_j^i(b_1^i)$ can never be empty since the true input sequence is always compatible with the observed output if the observed output is truly the output of a binary erasure channel. So, $\|\mathcal{C}_j^i(b_1^i)\|$, the cardinality of the set $\mathcal{C}_j^i(b_1^i)$, is bounded below by 1 while being monotonically decreasing. Thus, for any possible given realization of b_1^∞ we know $\lim_{i\to\infty} \|\mathcal{C}_j^i(b_1^i)\|$ exists and therefore so does the limiting set $\mathcal{C}_j^\infty(b_1^\infty)$.

In particular, if the set $\mathcal{C}_j^\infty(b_1^\infty)$ is a singleton, $\mathcal{C}_j^i(b_1^i)$ will have become a singleton at some finite time i. At that point, we know from the nesting properties that for all $k < j$, $\|\mathcal{C}_k^i(b_1^i)\| = 1$ as well. Finally, for convenience whenever $j \leq 0$, we set $\mathcal{C}_j^i = \{\emptyset\}$ so it is a singleton consisting only of the empty set.

Proposition 5.3.3 *For $0 < \lfloor(\theta + j)R\tau\rfloor \leq i$ we have:*

$$P(\|\mathcal{C}_j^i(B_1^i)\| > 1) \leq \sum_{k=1}^{\infty} P\left(\|\mathcal{C}_{j-k+1}^i(B_1^i)\| > 1 \,\middle|\, \|\mathcal{C}_{j-k}^i(B_1^i)\| = 1\right)$$

Proof:

$$\begin{aligned}
& P(\|\mathcal{C}_j^i(B_1^i)\| > 1) \\
= \;& P\left(\|\mathcal{C}_j^i(B_1^i)\| > 1 \,\middle|\, \|\mathcal{C}_{j-1}^i(B_1^i)\| = 1\right) P\left(\|\mathcal{C}_{j-1}^i(B_1^i)\| = 1\right) \\
& + P\left(\|\mathcal{C}_j^i(B_1^i)\| > 1 \,\middle|\, \|\mathcal{C}_{j-1}^i(B_1^i)\| > 1\right) P\left(\|\mathcal{C}_{j-1}^i(B_1^i)\| > 1\right) \\
= \;& P\left(\|\mathcal{C}_j^i(B_1^i)\| > 1 \,\middle|\, \|\mathcal{C}_{j-1}^i(B_1^i)\| = 1\right) P(\|\mathcal{C}_{j-1}^i(B_1^i)\| = 1) + P(\|\mathcal{C}_{j-1}^i(B_1^i)\| > 1) \\
= \;& \sum_{k=1}^{j} P\left(\|\mathcal{C}_{j-k+1}^i(B_1^i)\| > 1 \,\middle|\, \|\mathcal{C}_{j-k}^i(B_1^i)\| = 1\right) P(\mathcal{C}_{j-k}^i(B_1^i)\| = 1) \\
\leq \;& \sum_{k=1}^{j} P\left(\|\mathcal{C}_{j-k+1}^i(B_1^i)\| > 1 \,\middle|\, \|\mathcal{C}_{j-k}^i(B_1^i)\| = 1\right) \\
\leq \;& \sum_{k=1}^{\infty} \left(\|\mathcal{C}_{j-k+1}^i(B_1^i)\| > 1 \,\middle|\, \|\mathcal{C}_{j-k}^i(B_1^i)\| = 1\right)
\end{aligned}$$

where the first inequality comes from the fact that probabilities are bounded above by 1. □

We need to be able to say something about $P\left(\|\mathcal{C}_{j-k+1}^i(B_1^i)\| > 1 \,\middle|\, \|\mathcal{C}_{j-k}^i(B_1^i)\| = 1\right)$ in order to use this proposition. This is the probability that the first remaining ambiguity in the stream is seen at bit $j-k+1$, given that it is unambiguous before that point. To bound it, we will consider a random encoder $\mathcal{E}$ with $P(0) = P(1) = \frac{1}{2}$.

Proposition 5.3.4 *Assuming the input stream S_1^∞ is generated by i.i.d. fair coins independent of the rate R random encoder $\mathcal{E}$, for the binary erasure channel we have*

$$P(\|\mathcal{C}_{j-k+1}^i(B_1^i)\| > 1 \,|\, \|\mathcal{C}_{j-k}^i(B_1^i)\| = 1) \leq 2\left(\frac{1+e}{2^{1-R\tau}}\right)^{i-\lceil\frac{j-k+1-\theta}{R\tau}\rceil+1}$$

where the probability is taken over both the channel and the random ensemble of encoders.

Proof: For the event $\{\|\mathcal{C}_{j-k+1}^i(B_1^i)\| > 1 \,|\, \|\mathcal{C}_{j-k}^i(B_1^i)\| = 1\}$ to happen, there must be at least one sequence of bits $(\tilde{s}_{j-k+2}, \ldots, \tilde{s}_{\lfloor(\theta+i)R\tau\rfloor})$ such that the complete sequence $(s_1, \ldots, s_{j-k}, 1-s_{j-k+1}, \tilde{s}_{j-k+2}, \ldots, \tilde{s}_{\lfloor(\theta+i)R\tau\rfloor})$ is not incompatible with B_1^i. The first spot in the channel input where this would possibly be visible is $\left\lceil\frac{j-k+1-\theta}{R\tau}\right\rceil$, the index of the first channel input which depends on the erroneous bit $(1 - s_{j-k+1})$.

Let $\tilde{A}^l = (\tilde{A}_{\lceil\frac{j-k+1-\theta}{R\tau}\rceil}, \ldots, \tilde{A}_i)$ be a candidate transmitted string that, due to channel errors, ends up indistinguishable from the true transmitted string $A^l = (A_{\lceil\frac{j-k+1-\theta}{R\tau}\rceil}, \ldots, A_i)$, where l, the length of this sequence, is:

$$l = i - \left\lceil\frac{j-k+1-\theta}{R\tau}\right\rceil + 1 \tag{5.1}$$

Because of the erroneous bit $(1 - s_{j-k+1})$, we know by the definition of a random encoder that the $\tilde{A}^l$ is independent of the true A^l.

The probability that a randomly generated $\tilde{A}^l$ would be indistinguishable from a particular A^l is: $\sum_{m=0}^{l} \frac{\frac{1}{2^m} l! e^{l-m}(1-e)^m}{m!(l-m)!}$ where $l-m$ represents the number of channel errors that occur during the transmission of this sequence, $\frac{1}{2^m}$ is the probability of matching a random binary string of length m, and the other terms represent the probability of seeing $l-m$ errors in a sequence of length l where the error probability is e and errors are i.i.d. This probability sum can be simplified to $(\frac{1+e}{2})^l$ and represents the probability that a particular candidate $\tilde{A}^l$ string is not inconsistent.

To bound the probability that there exists at least one such candidate, we sum over all possible candidates. Regardless of offset θ, a sequence of l channel uses spans at most $\lceil lR\tau \rceil < 1 + lR\tau$ input bits and there are only at most $2^{1+lR\tau}$ possible bit sequences. So, $P\left(\|\mathcal{C}^i_{j-k+1}(B^i_1)\| > 1 \middle| \|\mathcal{C}^i_{j-k}(B^i_1)\| = 1\right) \leq 2^{1+lR\tau}(\frac{1+e}{2})^l = 2\left(\frac{1+e}{2^{1-R\tau}}\right)^l$ giving us the desired result when we substitute in the definition for l given in (5.1). $\square$

The approximations used in Proposition 5.3.4 above are conservative, since there are not really $2^{\lceil lR\tau \rceil}$ independent possibilities for $\tilde{A}^l$. There are that many choices for the final letter of the string, while there are at most two possibilities for the first letter.

Combining Propositions 5.3.3 and 5.3.4 gives us the following theorem.

Theorem 5.3.1 *For the binary erasure channel with erasure probability e,* $C_{anytime}(\alpha) \geq \frac{1-\log_2(1+e)}{\tau} - \alpha$ *with the inequality satisfied by a suitable rate encoder/decoder system with common randomness.*

Proof: At time i we have $P(\hat{S}_j \neq S_j) \leq P(\|\mathcal{C}^i_j(B^i_1)\| > 1)$ since there has to be some ambiguity if there is an error. Then we can use Propositions 5.3.3 and 5.3.4 to give us:

$$\begin{aligned}
P(\|\mathcal{C}^i_j(B^i_1)\| > 1) &\leq \sum_{k=1}^{\infty} P\left(\|\mathcal{C}^i_{j-k+1}(B^i_1)\| > 1 \middle| \|\mathcal{C}^i_{j-k}(B^i_1)\| = 1\right) \\
&< \sum_{k=1}^{\infty} 2\left(\frac{1+e}{2^{1-R\tau}}\right)^{i-\lceil \frac{j-k+1-\theta}{R\tau} \rceil + 1} \\
&< 2 \sum_{l=i-\lceil \frac{j-\theta}{R\tau} \rceil}^{\infty} \left(\frac{1+e}{2^{1-R\tau}}\right)^l \\
&= \frac{2\left(\frac{1+e}{2^{1-R\tau}}\right)^{i-\lceil \frac{j-\theta}{R\tau} \rceil}}{1-\frac{1+e}{2^{1-R\tau}}} \\
&\leq \frac{2\left(\frac{1+e}{2^{1-R\tau}}\right)^{-1+i-\frac{j-\theta}{R\tau}}}{1-\frac{1+e}{2^{1-R\tau}}}
\end{aligned}$$

The geometric sums only converge if $1 + e < 2^{1-R\tau}$ or equivalently $R < \frac{1-\log_2(1+e)}{\tau}$. In that case, notice that the probability of ambiguity at the j-th bit at time i is an exponentially decreasing function of the time delay $i\tau - \frac{j-\theta}{R}$. Taking logarithms and solving for the information rate R as a function of the exponential decay rate α gives us $R = \frac{1-\log_2(1+e)}{\tau} - \alpha$ proving the theorem. $\square$

This theorem establishes a lower bound on the anytime capacity of the binary erasure channel. It is easy to see that this lower bound does not conflict with the classical Shannon capacity of this channel $\frac{1-e}{\tau}$ since $R < \frac{1-\log_2(1+e)}{\tau} \leq \frac{1-e}{\tau}$. However, the way we proved it showed that for this particular channel, the anytime decoder can actually provide additional information. It can tell us when it knows the transmitted bit for sure (when $\mathcal{C}_j^i$ is a singleton) and when it is still uncertain. In this aspect, it is like the situation when using the system with feedback given in (2.2) and (2.3) that implicitly requests retransmission of garbled bits. There too the decoder can tell when it knows a bit correctly.

5.3.2 Additive White Gaussian Noise Channels

To see that the binary erasure channel is not a fluke, we examine the scalar additive white Gaussian noise channel with a power constraint P. Without loss of generality, let us adjust units so that the Gaussian noise on the channel has unit variance. Once again, we will use a random rate R encoder $\mathcal{E}$ with channel inputs drawn i.i.d. according to a zero-mean Gaussian with variance P. It should be clear how the encoder satisfies the power constraint for the channel by construction.

The main difference in the AWGN case is that there is nothing analogous to Proposition 5.3.1 since it is impossible to completely eliminate any possibility. But it will be possible to get an analog of Proposition 5.3.3.

First, let us look more closely at the maximum-likelihood anytime decoder. Because the received value is drawn from a continuous set, we must use a density rather than a probability.

$$\begin{aligned}
\mathcal{D}_i^a(b_1^i) &= \arg \max_{s_1^{\lfloor(\theta+i)R\tau\rfloor} \in \{0,1\}^{\lfloor(\theta+i)R\tau\rfloor}} p\left(B_1^i = b_1^i \,\middle|\, A_1^i = \mathcal{E}(s_1^{\lfloor(\theta+i)R\tau\rfloor})\right) \\
&= \arg \min_{s_1^{\lfloor(\theta+i)R\tau\rfloor} \in \{0,1\}^{\lfloor(\theta+i)R\tau\rfloor}} \sum_{j=1}^{i} (b_j - \mathcal{E}_j(s_1^{\lfloor(\theta+j)R\tau\rfloor}))^2 \\
&= \arg \min_{s_1^{\lfloor(\theta+i)R\tau\rfloor} \in \{0,1\}^{\lfloor(\theta+i)R\tau\rfloor}} \frac{1}{i}\sum_{j=1}^{i} (b_j - \mathcal{E}_j(s_1^{\lfloor(\theta+j)R\tau\rfloor}))^2
\end{aligned}$$

The intuition is that the true path will have an average cost near 1 since the cost will just be the result of the channel noise. Meanwhile, false paths will eventually have average costs near $1 + 2P$. If we wait long enough, the strong law of large numbers tells us that all false paths will reveal their falsity.

Proposition 5.3.5 *For* $0 < \lfloor(\theta + j)R\tau\rfloor \leq i$ *we have:*

$$P(\left(\mathcal{D}_i^a(B_1^i)\right)_j \neq s_j) \leq \sum_{k=1}^{\infty} P(\left(\mathcal{D}_i^a(B_1^i)\right)_{j-k+1} \neq s_{j-k+1} \,\middle|\, \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k})$$

Proof: This is very similar to Proposition 5.3.3 and proved by induction.

$$\begin{aligned}
& P(\left(\mathcal{D}_i^a(B_1^i)\right)_j \neq s_j) \\
\leq\ & 0P(\left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j} \neq s_{-\infty}^{j}) \\
=\ & P(\left(\mathcal{D}_i^a(B_1^i)\right)_j \neq s_j \,\middle|\, \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-1} = s_{-\infty}^{j-1})P(\left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-1} = s_{-\infty}^{j-1}) \\
& +P(\left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-1} \neq s_{-\infty}^{j-1})
\end{aligned}$$

$$
\begin{aligned}
&\le P(\left(\mathcal{D}_i^a(B_1^i)\right)_j \neq s_j \Big| \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-1} = s_{-\infty}^{j-1}) + P(\left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-1} \neq s_{-\infty}^{j-1}) \\
&\le \sum_{k=1}^{j} P(\left(\mathcal{D}_i^a(B_1^i)\right)_{j-k+1} \neq s_{j-k+1} \Big| \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k}) \\
&\le \sum_{k=1}^{\infty} P(\left(\mathcal{D}_i^a(B_1^i)\right)_{j-k+1} \neq s_{j-k+1} \Big| \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k})
\end{aligned}
$$

The base case of the implicit induction holds because by assumption all bits must agree at or before time zero. □

As before, in order to use this proposition, we need to be able to say something about the probability $P((\mathcal{D}_i^a(B_1^i))_{j-k+1} \neq s_{j-k+1} \big| (\mathcal{D}_i^a(B_1^i))_{-\infty}^{j-k} = s_{-\infty}^{j-k})$. This is the probability that our decoded stream has the first error at bit $j-k+1$, given that it does not have any errors before that point.

Proposition 5.3.6 *Assuming the input stream S_1^∞ is generated by i.i.d. fair coins independent of the rate R random encoder $\mathcal{E}$, for the additive white Gaussian noise channel we have $P((\mathcal{D}_i^a(B_1^i))_{j-k+1} \neq s_{j-k+1} \big| (\mathcal{D}_i^a(B_1^i))_{-\infty}^{j-k} = s_{-\infty}^{j-k}) \le 2\left(\frac{2^{R\tau}}{\sqrt{1+\frac{P}{2}}}\right)^{i-\lceil \frac{j-k+1-\theta}{R\tau} \rceil + 1}$ where the probability is over both the channel and the random ensemble of encoders.*

Proof: The proof of this is close to that of Proposition 5.3.4. For the event to happen, there must be a string of bits $(\tilde{s}_{j-k+2}, \ldots, \tilde{s}_{\lfloor(\theta+i)R\tau\rfloor})$ such that the complete sequence of bits $(s_1, \ldots, s_{j-k}, \bar{s}_{j-k+1}, \tilde{s}_{j-k+2}, \ldots, \tilde{s}_{\lfloor(\theta+i)R\tau\rfloor})$ has an encoding which is closer to B_1^i than the encoding of the true path. Clearly, the first point n where it is possible that $\tilde{A}_n \neq A_n$ is $n = \left\lceil \frac{j-k+1-\theta}{R\tau} \right\rceil$, the index of the first channel input which depends on the erroneous bit $\bar{s}_{j-k+1}$. Once again, we define l, the length of the sequence of channel transmissions after the erroneous bit:

$$
l = i - \left\lceil \frac{j-k+1-\theta}{R\tau} \right\rceil + 1 \tag{5.2}
$$

Then we have:

$$
\begin{aligned}
& P(\left(\mathcal{D}_i^a(B_1^i)\right)_{j-k+1} \neq s_{j-k+1} \Big| \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k}) \\
&= P(\exists \tilde{s}_{j-k+2}^{\lfloor(\theta+i)R\tau\rfloor} : \sum_{n=1}^{i} (B_n - A_n)^2 \ge \sum_{n=1}^{i} (B_n - \tilde{A}_n)^2 \Big| \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k}) \\
&= P(\exists \tilde{s}_{j-k+2}^{\lfloor(\theta+i)R\tau\rfloor} : \sum_{n=\lceil \frac{j-k+1-\theta}{R} \rceil}^{i} (B_n - A_n)^2 \ge \sum_{n=\lceil \frac{j-k+1-\theta}{R\tau} \rceil}^{i} (B_n - \tilde{A}_n)^2) \\
&= P(\exists \tilde{s}_{j-k+2}^{\lfloor(\theta+i)R\tau\rfloor} : \sum_{n=\lceil \frac{j-k+1-\theta}{R\tau} \rceil}^{i} (V_n)^2 \ge \sum_{n=\lceil \frac{j-k+1-\theta}{R\tau} \rceil}^{i} (V_n + A_n - \tilde{A}_n)^2)
\end{aligned}
$$

By the construction of our random encoders, the V_n, A_n, and $\tilde{A}_n$ in the final sums above are independent of each other. So we consider a random candidate $\tilde{A}^l$ and calculate:

$$
P(\sum_{n=1}^{l} (V_n)^2 \ge \sum_{n=1}^{l} (V_n + A_n - \tilde{A}_n)^2) \quad = \quad P(\sum_{n=1}^{l} (V_n + A_n - \tilde{A}_n)^2) - V_n^2 \le 0)
$$

$$
\begin{aligned}
&= P(\sum_{n=1}^{l}(A_n - \tilde{A}_n)^2 + 2V_n(A_n - \tilde{A}_n) \leq 0) \\
&= P(\sum_{n=1}^{l} 2\sqrt{2P}V_n Z_n - 2PZ_n^2 \geq 0) \\
&= P(\sum_{n=1}^{l} V_n Z_n - \sqrt{\frac{P}{2}}Z_n^2 \geq 0)
\end{aligned}
$$

where Z_n is an i.i.d. unit variance Gaussian like V_n. To bound this probability, we will use the Chernoff Bound:

$$
\begin{aligned}
P(\sum_{n=1}^{l}(V_n)^2 \geq \sum_{n=1}^{l}(V_n + A_n - \tilde{A}_n)^2) &\leq \min_{t\geq 0} E\left[e^{t(\sum_{n=1}^{l}(V_n Z_n - \sqrt{\frac{P}{2}}Z_n^2))}\right] \\
&= \min_{t\geq 0} E\left[\prod_{n=1}^{l} e^{t(V_n Z_n - \sqrt{\frac{P}{2}}Z_n^2)}\right] \\
&= \min_{t\geq 0} \prod_{n=1}^{l} E\left[e^{t(V_n Z_n - \sqrt{\frac{P}{2}}Z_n^2)}\right] \\
&= \min_{t\geq 0}(E\left[e^{t(VZ - \sqrt{\frac{P}{2}}Z^2)}\right])^l \\
&= \left(\min_{t\geq 0}\int\int e^{tvz - t\sqrt{\frac{P}{2}}z^2}\frac{1}{2\pi}e^{-\frac{v^2+z^2}{2}}dvdz\right)^l \\
&= \left(\min_{t\geq 0}\frac{1}{\sqrt{2\pi}}\int e^{(\frac{t^2-1}{2} - t\sqrt{\frac{P}{2}})z^2}dz\right)^l \\
&= \left(\min_{t\geq 0}\frac{1}{\sqrt{1 + t(\sqrt{2P} - t)}}\right)^l
\end{aligned}
$$

The final integral only converges to that value if $t^2 - t\sqrt{2P} - 1 < 0$ and is infinite otherwise. That does not effect the minimization, which is achieved at $t = \sqrt{\frac{P}{2}}$ well within the convergence region. Plugging this value back in gives us:

$$
P(\sum_{n=1}^{l}(V_n)^2 \geq \sum_{n=1}^{l}(V_n + A_n - \tilde{A}_n)^2) \leq \left(\frac{1}{\sqrt{1 + \frac{P}{2}}}\right)^l
$$

To bound the probability that there exists at least one such candidate, we sum over all possible candidates. Regardless of the offset θ, a sequence of l channel uses spans at most $\lceil lR\tau \rceil < 1 + lR\tau$ input bits and there are only upto $2^{1+lR\tau}$ possible bit sequences. So:

$$
\begin{aligned}
&P(\left(\mathcal{D}_i^a(B_1^i)\right)_{j-k+1} \neq s_{j-k+1} \middle| \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k}) \\
\leq\ & P\left(\exists \tilde{s}_{j-k+2}^{\lfloor(\theta+i)R\tau\rfloor} : \sum_{n=\lceil\frac{j-k+1-\theta}{R\tau}\rceil}^{i}(V_n)^2 \geq \sum_{n=\lceil\frac{j-k+1-\theta}{R\tau}\rceil}^{i}(V_n + A_n - \tilde{A}_n)^2\right)
\end{aligned}
$$

$$\leq \ 2^{1+lR\tau}\left(\frac{1}{\sqrt{1+\frac{P}{2}}}\right)^{l}$$

$$= \ 2\left(\frac{2^{R\tau}}{\sqrt{1+\frac{P}{2}}}\right)^{l}$$

By substituting the definition for l from (5.2), we have the desired result. □

Once again, the approximations used in Proposition 5.3.6 above are conservative for the same reason as in Proposition 5.3.4.

Theorem 5.3.2 *For the additive white Gaussian noise channel with power constraint P, $C_{anytime}(\alpha) \geq \frac{1}{2\tau}\log_2(1+\frac{P}{2})-\alpha$ with the inequality achieved by an encoder/decoder system with common randomness.*

Proof: Propositions 5.3.5 and 5.3.6 combine to give us:

$$\begin{aligned}
P(\left(\mathcal{D}_i^a(B_1^i)\right)_j \neq s_j) &\leq \sum_{k=1}^{\infty} P(\left(\mathcal{D}_i^a(B_1^i)\right)_{j-k+1} \neq s_{j-k+1} \,\Big|\, \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k}) \\
&\leq \sum_{k=1}^{\infty} 2\left(\frac{2^{R\tau}}{\sqrt{1+\frac{P}{2}}}\right)^{i-\lceil\frac{j-k+1-\theta}{R\tau}\rceil+1} \\
&\leq \max(1,R\tau) \sum_{l=i-\lceil\frac{j-k+1-\theta}{R\tau}\rceil+1}^{\infty} 2\left(\frac{2^{R\tau}}{\sqrt{1+\frac{P}{2}}}\right)^{l} \\
&= \frac{2\max(1,R\tau)}{1-\frac{2^{R\tau}}{\sqrt{1+\frac{P}{2}}}}\left(\frac{2^{R\tau}}{\sqrt{1+\frac{P}{2}}}\right)^{i-\lceil\frac{j-k+1-\theta}{R\tau}\rceil+1} \\
&\leq \frac{2\max(1,R\tau)}{1-\frac{2^{R\tau}}{\sqrt{1+\frac{P}{2}}}}\left(\frac{2^{R\tau}}{\sqrt{1+\frac{P}{2}}}\right)^{i-\frac{j-k+1-\theta}{R\tau}}
\end{aligned}$$

The geometric sums only converge if $2^{R\tau} < \sqrt{1+\frac{P}{2}}$ which means $R < \frac{1}{2\tau}\log_2(1+\frac{P}{2})$. In that case, notice that the probability of error at the j-th bit at time i is an exponentially decreasing function of the delay $i\tau - \frac{j}{R}$. Taking logs and solving for the information rate R as a function of the exponential decay rate α gives us $R = \frac{1}{2\tau}\log_2(1+\frac{P}{2}) - \alpha$ proving the theorem. □

So, analogous to Theorem 5.3.1, this theorem establishes a lower bound on the anytime capacity of the AWGN channel. It is easy to see that this lower bound does not conflict with the classical Shannon capacity of this channel $\frac{1}{2\tau}\log_2(1+P)$ since $R < \frac{1}{2\tau}\log_2(1+\frac{P}{2}) < \frac{1}{2\tau}\log_2(1+P)$.

5.3.3 The General Case

The erasure case was clearly special and was chosen because it is intuitively the easiest to understand. The AWGN case showed that anytime capacity was not restricted to the erasure case and the similarities with the erasure case also point the way to a more general treatment. In this section, we show how the same basic argument extends to general memoryless channels.

In the proofs for both the erasure channel and the AWGN channel, the key role was played by Propositions 5.3.3 and 5.3.5 respectively. Both of these allowed us to bound the probability of error on any individual bit as the infinite sum of probability of errors on longer strings. This then allowed us to use Propositions 5.3.4 and 5.3.6 which gave us exponential decaying upper bounds for the longer and longer strings. Because the sum of exponentials is bounded, this allowed us to straightforwardly prove Theorems 5.3.1 and 5.3.2.

It turns out that Propositions 5.3.3 and 5.3.5 have natural generalizations to any discrete time channel for which we can represent the maximum likelihood decoder as something which minimizes a cost function between the received signal and a supposed transmitted one. The property we need is just that this function should be positive and additive in the following way:

Definition 5.3.3 *A family of positive functions* $\xi_i : A^i \times B^i \to \Re_+ \cup \{+\infty\}$ *is a* decoding cost function *if* $\forall i, j > 0$ *we have:*

$$\xi_{i+j}(a_1^{i+j}, b_1^{i+j}) = \xi_i(a_1^i, b_1^i) + \xi_j(a_{i+1}^{i+j}, b_{i+1}^{i+j}) = \sum_{k=1}^{i+j} \xi(a_k, b_k)$$

for all strings a_1^∞ *and* b_1^∞.

The decoding cost function ξ *is used to determine an anytime* ξ*-decoder for an* (R, τ, θ) *code as follows:*

$$\mathcal{D}_i^\xi(b_1^i) = \arg \min_{s_1^{\lfloor(\theta+i)R\tau\rfloor} \in \{0,1\}^{\lfloor(\theta+i)R\tau\rfloor}} \xi_i(\mathcal{E}(s_1^{\lfloor(\theta+i)R\tau\rfloor}), b_1^i)$$

It should be easy to see that for discrete memoryless channels, maximum likelihood decoding is the same as minimizing the decoding cost function where $\xi(a,b) = -\log p(b|a)$. $-\log 0 = +\infty$ is used to represent the cost if b cannot occur if a is input to the channel. This is because log is a monotonic function and a memoryless channel has transition probabilities which are products and get turned into sums by the log function.

For such decoding cost functions, we can easily prove the following generalization of Propositions 5.3.3 and 5.3.5:

Lemma 5.3.1 *For* $0 < \lfloor(\theta + j)R\tau\rfloor \leq i$ *we have:*

$$P(\left(\mathcal{D}_i^\xi(B_1^i)\right)_j \neq s_j) \leq \sum_{k=1}^{\infty} P(\left(\mathcal{D}_i^\xi(B_1^i)\right)_{j-k+1} \neq s_{j-k+1} \Big| \left(\mathcal{D}_i^\xi(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k})$$

Proof: The proof is identical to that of Proposition 5.3.5. □

We can also establish the following Lemma which serves as the analogue of Propositions 5.3.3 and 5.3.5. It effectively relates the error exponent for independently generated random block codes to what we need.

Lemma 5.3.2 *Let* $P(a)$ *be the distribution with which our random encoder has been generated. Consider a block-coding situation where we randomly generate a correct codeword of length* N *using* $P(a)$ *and then randomly generate* $2^{NR} - 1 \leq M - 1 < 2^{NR}$ *incorrect codewords using the same distribution with each of the symbols in the incorrect codewords being* *independent of the true codeword. If under the decoding metric ξ for all N, the probability of error is less than $K2^{-NK'}$, then the anytime ξ-decoder for our random encoder has*

$$P(\left(\mathcal{D}_i^a(B_1^i)\right)_{j-k+1} \neq s_{j-k+1} \bigg| \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k}) \leq K2^{-K'(i-\lceil \frac{j-k+1-\theta}{R\tau}\rceil+1)}$$

where the probability is over both the channel and the random ensemble of encoders.

Proof: The proof follows Proposition 5.3.6 closely. For the error event:

$$\{\left(\mathcal{D}_i^a(B_1^i)\right)_{j-k+1} \neq s_{j-k+1} \bigg| \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k}\}$$

to happen, there must be a string of bits $(\tilde{s}_{j-k+2}, \ldots, \tilde{s}_{\lfloor(\theta+i)R\tau\rfloor})$ such that

$$(s_1, \ldots, s_{j-k}, \bar{s}_{j-k+1}, \tilde{s}_{j-k+2}, \ldots, \tilde{s}_{\lfloor(\theta+i)R\tau\rfloor})$$

has an encoding which is closer to B_1^i than the encoding of the true path is under the ξ decoding metric. Clearly, the first position where it is possible that $\tilde{A}_n \neq A_n$ is $n = \left\lceil \frac{j-k+1-\theta}{R\tau} \right\rceil$, the index of the first channel input which depends on the erroneous bit $\bar{s}_{j-k+1}$. Once again, we define l, the length of the sequence of channel transmissions after the erroneous bit:

$$l = i - \left\lceil \frac{j-k+1-\theta}{R\tau} \right\rceil + 1 \tag{5.3}$$

Then we have

$$\begin{aligned}
& P(\left(\mathcal{D}_i^a(B_1^i)\right)_{j-k+1} \neq s_{j-k+1} \bigg| \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k}) \\
= \; & P(\exists \tilde{s}_{j-k+2}^{\lfloor(\theta+i)R\tau\rfloor} : \sum_{n=1}^{i} \xi(A_n, B_n) \geq \sum_{n=1}^{i} \xi(\tilde{A}_n, B_n) \bigg| \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k}) \\
= \; & P(\exists \tilde{s}_{j-k+2}^{\lfloor(\theta+i)R\tau\rfloor} : \sum_{n=\lceil \frac{j-k+1-\theta}{R}\rceil}^{i} \xi(A_n, B_n) \geq \sum_{n=\lceil \frac{j-k+1-\theta}{R\tau}\rceil}^{i} \xi(\tilde{A}_n, B_n)) \\
\leq \; & K2^{-K'l}
\end{aligned}$$

The final inequality comes from the fact that all the channel inputs corresponding to the false string are pairwise independent of the true ones under our definition of random encoder. Their length is l and hence they can be considered as the block case above. Plugging in our definition for l in (5.3) gives the Lemma. □

Finally, we can put Lemmas 5.3.1 and 5.3.2 together with Gallager's $E_r(R)$ standard block random coding exponent [23] to get the following theorem:

Theorem 5.3.3 *For a memoryless channel with block random coding exponent $E_r(R)$, the $C_{anytime}(E_r(R)\log_2 e) \geq R\log_2 e$ with the inequality satisfied by a suitable rate encoder/decoder system with common randomness.*

Proof: Recall that there exists a distribution $P(a)$ using which we can make a block-code by drawing $e^{NR} = 2^{NR\log_2 e}$ length N codewords according to $P(a)$ on each letter. The expected probability of error is less than or equal to $e^{-NE_r(R)} = 2^{-N(E_r(R)\log_2 e)}$ where $E_r(R)$ is Gallager's random coding exponent. We now generate a random code in our sense using the same $P(a)$ and then apply the combination of Lemmas 5.3.1 and 5.3.2 to get:

$$\begin{aligned}
P(\left(\mathcal{D}_i^a(B_1^i)\right)_j \neq s_j) &\leq \sum_{k=1}^{\infty} P(\left(\mathcal{D}_i^a(B_1^i)\right)_{j-k+1} \neq s_{j-k+1} \Big| \left(\mathcal{D}_i^a(B_1^i)\right)_{-\infty}^{j-k} = s_{-\infty}^{j-k}) \\
&\leq \sum_{k=1}^{\infty} \left(2^{-(E_r(R)\log_2 e)}\right)^{i-\left\lceil \frac{j-k+1-\theta}{\tau R \log_2 e} \right\rceil + 1} \\
&\leq \max(1, \tau R \log_2 e) \sum_{l=i-\left\lceil \frac{j-k+1-\theta}{\tau R \log_2 e} \right\rceil + 1}^{\infty} \left(2^{-(E_r(R)\log_2 e)}\right)^{l} \\
&= \frac{\max(1, \tau R \log_2 e)}{1 - 2^{-(E_r(R)\log_2 e)}} \left(2^{-(E_r(R)\log_2 e)}\right)^{i-\left\lceil \frac{j-k+1-\theta}{\tau R \log_2 e} \right\rceil + 1} \\
&\leq \frac{\max(1, \tau R \log_2 e)}{1 - 2^{-(E_r(R)\log_2 e)}} \left(2^{-(E_r(R)\log_2 e)}\right)^{i-\frac{j-k+1-\theta}{\tau R \log_2 e}}
\end{aligned}$$

The geometric sums always converge if $E_r(R) > 0$. This proves the theorem. □

In Theorem 5.3.3, the $\log_2 e$ terms exist only because Gallager's block random coding exponent is defined in terms of nats and base e while we are using bits and base 2. It should be clear that Theorems 5.3.1 and 5.3.2 can be tightened up just by using the tighter $E_r(R)$ estimates instead of our loose ones. Because the block random coding exponent has been shown to be positive for all rates below Shannon's classical capacity, we also know that the 0-anytime capacity (thought of as a limit where the exponent tends to zero) is equal to Shannon's classical capacity!

5.4 From Random to Deterministic Codes

The proof of Theorem 5.3.3 establishes that with common randomness, we can have a non-zero α-anytime capacity even without any feedback. As we know, common randomness can be interpreted as a measure over the space of deterministic codes. The performance for a code with common randomness is really the expected performance of deterministic encoders with regard to that measure. It would be nice to be able to show the existence of a deterministic code with the required property by itself.

In standard random-coding arguments for Shannon capacity and block codes, this argument is very simple to make. The probability measure is over a reasonably small set and the notion of reliability is also finite. Hence it is easy to see that at least one member of the set does at least as well as the expectation over the whole set. In contrast, our probability measure is over an uncountable number of infinite encoders being evaluated on a sense of reliability that also has an infinite dimension. The three line arguments no longer apply.

Since all of our random coding results are in terms of exponentials, we will focus only on the exponentially decaying case in this section. First, we notice:

Lemma 5.4.1 *Let $\{\mathcal{E}\}$ be the set of deterministic rate R, offset θ encoders with some probability measure P over them. If $E_{\mathcal{E}}\left[P_{error}(\mathcal{E}, \mathcal{D}^a, d, i)\right]$ decays exponentially in d with asymptotic exponent at least α, then so does the expectation over any set of encoders with strictly positive probability. The same also applies to their supremum $P_{error}(\mathcal{E}, \mathcal{D}^a, d_1^{\infty} = d)$.*

Proof: We will prove the following equivalent statement: If $\mathcal{Q} \subseteq \{\mathcal{E}\}$ has $P(\mathcal{Q}) > 0$ and

$$E_{\mathcal{E}\in\mathcal{Q}}\left[P_{\text{error}}(\mathcal{E},\mathcal{D}^a,d,i)\right] \geq K_i 2^{-\alpha_1 d}$$

for $d > 0$, then $E_{\mathcal{E}\in\{\mathcal{E}\}}\left[P_{\text{error}}(\mathcal{E},\mathcal{D}^a,d,i)\right] \geq K_i' 2^{-\alpha_1 d}$ as well.

$$\begin{aligned}
& E_{\mathcal{E}\in\{\mathcal{E}\}}\left[P_{\text{error}}(\mathcal{E},\mathcal{D}^a,d,i)\right] \\
= \; & P(\mathcal{Q})E_{\mathcal{E}\in\mathcal{Q}}\left[P_{\text{error}}(\mathcal{E},\mathcal{D}^a,d,i)\right] + (1-P(\mathcal{Q}))E_{\mathcal{E}\in\overline{\mathcal{Q}}}\left[P_{\text{error}}(\mathcal{E},\mathcal{D}^a,d,i)\right] \\
\geq \; & P(\mathcal{Q})E_{\mathcal{E}\in\mathcal{Q}}\left[P_{\text{error}}(\mathcal{E},\mathcal{D}^a,d,i)\right] \\
\geq \; & P(\mathcal{Q})K_i 2^{-\alpha_1 d}
\end{aligned}$$

So, $K_i' = P(\mathcal{Q})K_i > 0$, establishing the statement. This means that no positive measure subset of $\{\mathcal{E}\}$ can have a worse expected asymptotic exponent than the expectation over the whole set for any time t and thus also for their supremum. □

Lemma 5.4.1 only establishes that for all i, all positive measure sets of encoders (when viewed under the probability measure induced by our random coding procedure) have expected error probabilities which decay exponentially with delay. Furthermore the rate of decay is at least as good as the random encoder. All this tells us is that there cannot be any large sets of "bad" encoders all of which are bad in the same places. But we can say more about the behavior of individual deterministic encoders.

Lemma 5.4.2 *Let $\{\mathcal{E}\}$ be the set of deterministic rate R, offset θ encoders with some probability measure P over them and suppose $E_{\mathcal{E}}\left[P_{error}(\mathcal{E},\mathcal{D}^a,d,i)\right] \leq K_i 2^{-\alpha d}$ for all $d \geq 0$. Then for every $\epsilon > 0$, almost every deterministic encoder $\mathcal{E}$ has a constant $K_i^{\epsilon,\mathcal{E}} > 0$ such that $P_{error}(\mathcal{E},\mathcal{D}^a,d,i) \leq K_i^{\epsilon,\mathcal{E}} 2^{-(\alpha-\epsilon)d}$. Furthermore, the probability (over random encoders) $P(K_i^{\epsilon,\mathcal{E}} > K)$ decays at least as fast as $\frac{1}{K^{\frac{\alpha}{\alpha-\epsilon}}}$.*

Proof: For all $\epsilon > 0$ and $K > 0$, let $\mathcal{Q}^d_{(\epsilon,K,i)}$ be the set of encoders $\mathcal{E}$ with $P_{\text{error}}(\mathcal{E},\mathcal{D}^a,d,i) \geq K 2^{-(\alpha-\epsilon)d}$. Since probabilities can never exceed 1, we know that $\mathcal{Q}^d_{(\epsilon,K,i)}$ is empty whenever $K > 2^{(\alpha-\epsilon)d}$. Hence, $P(\mathcal{Q}^d_{(\epsilon,K,i)}) = 0$ whenever $d < \frac{\log_2 K}{\alpha-\epsilon}$. Furthermore, because we know $E_{\mathcal{E}}\left[P_{\text{error}}(\mathcal{E},\mathcal{D}^a,d,i)\right] \leq K_i 2^{-\alpha d}$ for all $d > 0$, we also have by Markov's inequality:

$$\begin{aligned}
P(\mathcal{Q}^d_{(\epsilon,K,i)}) &\leq \frac{K_i 2^{-\alpha d}}{K 2^{-(\alpha-\epsilon)d}} \\
&\leq \frac{K_i}{K} 2^{-\epsilon d}
\end{aligned}$$

Now let $\mathcal{Q}_{(\epsilon,K,i)} = \bigcup_{d\geq 0} \mathcal{Q}^d_{\epsilon,K,i}$ represent the set of encoders whose probability of error on bit i ever goes above $K 2^{-(\alpha-\epsilon)d}$.

Recall that the only time delays that we need to consider are those which are integer multiples of the channel sampling time. Therefore, we have (for $K > 1$):

$$\begin{aligned}
P(\mathcal{Q}_{\epsilon,K}) &\leq \sum_{j=0}^{\infty} P(\mathcal{Q}^{j\tau}_{\epsilon,K}) \\
&= \sum_{j=\lceil \frac{\log_2 K}{(\alpha-\epsilon)\tau} \rceil}^{\infty} P(\mathcal{Q}^{j\tau}_{\epsilon,K})
\end{aligned}$$

$$
\begin{aligned}
&\le \sum_{j=\left\lceil \frac{\log_2 K}{(\alpha-\epsilon)\tau} \right\rceil}^{\infty} \frac{K_i}{K} 2^{-\epsilon j \tau} \\
&= \frac{1}{K} \frac{K_i 2^{-\epsilon \left\lceil \frac{\log_2 K}{(\alpha-\epsilon)\tau} \right\rceil \tau}}{1-2^{-\epsilon\tau}} \\
&\le \frac{1}{K} \frac{K_i 2^{-\epsilon\left(\frac{\log_2 K}{\alpha-\epsilon}\right)}}{1-2^{-\epsilon\tau}} \\
&= \frac{2^{-\frac{\epsilon}{\alpha-\epsilon}\log_2 K}}{K} \frac{K_i}{1-2^{-\epsilon\tau}} \\
&= \left(\frac{K_i}{1-2^{-\epsilon\tau}}\right) \frac{1}{K^{\frac{\alpha}{\alpha-\epsilon}}}
\end{aligned}
$$

This can be made arbitrarily small by choosing K sufficiently large. Thus for any $\epsilon > 0$ almost all encoders $\mathcal{E}$ have a $K_i^{\epsilon,\mathcal{E}}$ such that $P_{\text{error}}(\mathcal{E}, \mathcal{D}^a, d, i) \le K_i^{\epsilon,\mathcal{E}} 2^{-(\alpha-\epsilon)d}$ for all $d > 0$. □

Suppose that we further know (as we do from Theorem 5.3.3) that there exists a single K such that $E_{\mathcal{E}}\left[P_{\text{error}}(\mathcal{E}, \mathcal{D}^a, d, i)\right] \le K 2^{-\alpha d}$ for all $d \ge 0$ and $i \ge 0$. Then we can apply Lemma 5.4.2 to each i and take the intersection of a countable number of measure 1 sets to see that for all $\epsilon > 0$, almost every deterministic encoder $\mathcal{E}$ has an infinite sequence of constants $[K_1^{\epsilon,\mathcal{E}}, K_2^{\epsilon,\mathcal{E}}, \ldots, K_i^{\epsilon,\mathcal{E}}, \ldots]$. However, this is not enough because this sequence might always be unbounded. To get an anytime capacity with a deterministic encoder, we need a uniform (over i) bound on the constants $K_i^{\epsilon,\mathcal{E}}$.

Theorem 5.4.1 *Let $\{\mathcal{E}\}$ be the set of deterministic rate R, offset θ encoders with the probability measure P over them generated by a random coding procedure of definition 5.3.1. Suppose there exists $K > 0$ such that $E_{\mathcal{E}}\left[P_{error}(\mathcal{E}, \mathcal{D}^a, d, i)\right] \le K 2^{-\alpha d}$ for all $d \ge 0$ and all $i \ge 0$ and all offsets θ. Then, for almost every deterministic encoder $\mathcal{E}$, for every $\epsilon > 0$, there exists a $K^{\mathcal{E}}$ such that $P_{error}(\mathcal{E}, \mathcal{D}^a, d, i) \le K^{\mathcal{E}} 2^{-(\alpha-\epsilon)d}$ for all $d \ge 0$ and $\forall i \ge 0$.*

Proof: Consider a rate R encoder $\mathcal{E}$. Since it is an infinite sequence of encoding functions, we can denote by $\mathcal{E}^{s_1^j}$ the sequence of encoding functions beginning after bits s_1^j have already been received at the encoder. This sequence can itself be viewed as another rate R encoder, albeit one with another offset $\theta' = \left\lfloor \frac{j}{R\tau} - \theta \right\rfloor + \theta - \frac{j}{R\tau}$. To make things explicit, we can define $\mathcal{E}_k^{s_1^j}(\tilde{s}_1^{\lfloor(\theta'+k)R\tau\rfloor}) = \mathcal{E}_{k+\lfloor \frac{j}{R\tau} - \theta \rfloor}(s_1^j, \tilde{s}_1^{\lfloor(\theta'+k)R\tau\rfloor})$.

Furthermore, because of how we have defined our random encoders, if $s_1^j \ne \tilde{s}_1^j$ then $\mathcal{E}^{s_1^j}$ and $\mathcal{E}^{\tilde{s}_1^j}$ are independent and are drawn from the same distribution as the original encoders, except with the new offset θ'.

To be able to use Lemma 5.4.2 on these θ' offset encoders, we first pick an $0 < \epsilon < \frac{\alpha}{2}$. Then for all i, we have a single constant $\mu > 0$ such that $P(K_i^{\mathcal{E}} > K) \le \frac{\mu}{K^{\frac{\alpha}{\alpha-\epsilon}}} = \mu K^{1+\frac{\epsilon}{\alpha-\epsilon}}$. By Corollary B.2.1, this clearly converges fast enough to show that K_i (viewed as a random variable depending on the random encoder $\mathcal{E}$) has a finite mean $\bar{K}$ and we can also consider averages of independent samples of K_i.

For any given encoder $\mathcal{E}$, we want to consider the probability $P_{\text{anyerr}}(\mathcal{E}, \mathcal{D}^a, i, d)$ of the event that with a delay of d units after bit i our decoder makes an error on any of the bits up-to and including bit i. Mathematically, we have:

$$P_{\text{anyerr}}(\mathcal{E}, \mathcal{D}^a, i, d) = E_{\{S_1^{\lfloor(\theta+j)R\tau\rfloor}\}}\left[P(\exists l \leq i, S_l \neq (\mathcal{D}_j^a(B_1^j))_l)\right]$$

where $j = \left\lceil \frac{(i-\theta)}{R\tau} + \frac{d}{\tau} \right\rceil$ represents the channel transmission index at which we are decoding (d units after bit i arrives). We will now show by induction that there always exists positive $M_i^{\mathcal{E}}$ such that $P_{\text{anyerr}}(\mathcal{E}, \mathcal{D}^a, i, d) \leq M_i^{\mathcal{E}} 2^{-(\alpha-\epsilon)d}$. For the base case, by Lemma 5.4.2, we will just set $M_1^{\mathcal{E}} = K_1^{\mathcal{E}}$ and use the following to get the rest:

$$\begin{aligned} P_{\text{anyerr}}(\mathcal{E}, \mathcal{D}^a, i+\delta, d) &\leq P_{\text{anyerr}}(\mathcal{E}, \mathcal{D}^a, i, d + \frac{\delta}{R}) \\ &\quad + E_{\{S_1^{\lfloor(\theta+j)R\tau\rfloor}\}}\left[P(\exists l \leq i+\delta : S_l \neq (\mathcal{D}_j^a(B_1^j))_l \left| (\mathcal{D}_j^a(B_1^j))_1^i = S_1^i)\right.\right] \end{aligned}$$

Pick the smallest integer $\delta > 0$ so that $\frac{\delta}{R} > \tau$. (e.g. if $R\tau < 1$, then $\delta = 1$) Notice that the second term (with the expectation) is upper bounded by the probability of an error within the first δ bits of using the encoder $\mathcal{E}^{S_1^i}$. Applying the induction hypothesis and this idea of subencoders gives us:

$$\begin{aligned} P_{\text{anyerr}}(\mathcal{E}, \mathcal{D}^a, i+\delta, d) &\leq M_i^{\mathcal{E}} 2^{-(\alpha-\epsilon)(d+\frac{\delta}{R})} + E_{\{S_1^i\}}\left[\sum_{l=1}^{\delta} P_{\text{error}}(\mathcal{E}^{S_1^i}, \mathcal{D}^a, d, l)\right] \\ &\leq 2^{-\frac{(\alpha-\epsilon)\delta}{R}} M_i^{\mathcal{E}} 2^{-(\alpha-\epsilon)d} + E_{\{S_1^i\}}\left[\sum_{l=1}^{\delta} K_l^{\mathcal{E}^{S_1^i}} 2^{-(\alpha-\epsilon)d}\right] \\ &= \left(2^{-\frac{(\alpha-\epsilon)\delta}{R}} M_t^{\mathcal{E}} + \sum_{l=1}^{\delta} E_{\{S_1^i\}}\left[K_l^{\mathcal{E}^{S_1^i}}\right]\right) 2^{-(\alpha-\epsilon)d} \end{aligned}$$

Which allows us to define the following recurrence for almost every encoder $\mathcal{E}$:

$$\begin{aligned} M_{i+\delta}^{\mathcal{E}} &= 2^{-\frac{(\alpha-\epsilon)\delta}{R}} M_i^{\mathcal{E}} + \sum_{l=1}^{\delta} E_{\{S_1^i\}}\left[K_l^{\mathcal{E}^{S_1^i}}\right] \\ &= 2^{-\frac{(\alpha-\epsilon)\delta}{R}} M_i^{\mathcal{E}} + \sum_{l=1}^{\delta} \frac{1}{2^i} \sum_{S_1^i \in \{0,1\}^i} K_l^{\mathcal{E}^{S_1^i}} \end{aligned}$$

The idea is that since the $\mathcal{E}^{S_1^i}$ are independent of each other, the average $\frac{1}{2^i}\sum_{S_1^i} K_l^{\mathcal{E}^{S_1^i}}$ should tend towards $\bar{K}$. Furthermore, we can use our generalization of Chebychev's inequality (Corollary B.2.1) to see that $\exists B > 0$ such that: $P(\frac{1}{2^i}\sum_{S_1^i} K_l^{\mathcal{E}^{S_1^i}} \geq \bar{K}+\gamma) \leq \frac{B}{\gamma^{1+\frac{\epsilon}{\alpha-\epsilon}}} 2^{-\frac{\epsilon i}{\alpha-\epsilon}}$.

Let $\mathcal{T}_i^\gamma$ be the set of all rate R encoders $\mathcal{E}$ for which $\sum_{l=1}^{\delta} \frac{1}{2^i}\sum_{S_1^i} K_l^{\mathcal{E}^{S_1^i}} \geq \delta(\bar{K}+\gamma)$. Take unions to get $\mathcal{T}^\gamma = \bigcup_{i\geq 1} \mathcal{T}_i^\gamma$. Now, consider the probability of the complement of $\mathcal{T}^\gamma$:

$$\begin{aligned} P(\overline{\mathcal{T}^\gamma}) &= 1 - P(\mathcal{T}^\gamma) \\ &\geq 1 - \sum_{i=1}^{\infty} P(\mathcal{T}_i^\gamma) \end{aligned}$$

$$\begin{aligned}
&\geq 1-\sum_{i=1}^{\infty}\sum_{l=1}^{\delta} P(\frac{1}{2^i}\sum_{S_1^i} K_l^{\mathcal{E}^{S_1^i}} \geq \bar{K}+\gamma) \\
&\geq 1-\sum_{i=1}^{\infty}\delta\frac{B}{\gamma^{1+\frac{\epsilon}{\alpha-\epsilon}}}2^{-\frac{\epsilon i}{\alpha-\epsilon}} \\
&= 1-\frac{\delta B}{\gamma^{1+\frac{\epsilon}{\alpha-\epsilon}}}\sum_{i=1}^{\infty}2^{-\frac{\epsilon i}{\alpha-\epsilon}} \\
&= 1-\frac{\delta B}{2^{\frac{\epsilon i}{\alpha-\epsilon}}\gamma^{1+\frac{\epsilon}{\alpha-\epsilon}}}
\end{aligned}$$

By choosing γ large enough, this can be made arbitrarily close to 1. So we define $\overline{\mathcal{T}} = \bigcup_{\gamma\geq 0}\overline{\mathcal{T}^{\gamma}}$. It is clear that:

$$\begin{aligned}
P(\overline{\mathcal{T}}) &= P(\bigcup_{\gamma\geq 0}\overline{\mathcal{T}^{\gamma}}) \\
&\geq \sup_{\gamma\geq 0}(1-\frac{\delta B}{2^{\frac{\epsilon i}{\alpha-\epsilon}}\gamma^{1+\frac{\epsilon}{\alpha-\epsilon}}}) \\
&= 1
\end{aligned}$$

$\overline{\mathcal{T}^{\gamma}}$ represents all those encoders $\mathcal{E}$ for which $\sum_{l=1}^{\delta}\frac{1}{2^i}\sum_{S_1^i} K_l^{\mathcal{E}^{S_1^i}}$ never exceeds $\delta(\bar{K}+\gamma)$ and hence $M_i^{\mathcal{E}}$ never exceeds $\frac{\delta(\bar{K}+\gamma)}{1-2^{-\frac{(\alpha-\epsilon)\delta}{R}}}$. Therefore, almost every encoder $\mathcal{E}\in\overline{\mathcal{T}}$ has a finite $M^{\mathcal{E}}>0$ such that $P_{\text{anyerr}}(\mathcal{E},\mathcal{D}^a,i,d)\leq M^{\mathcal{E}}2^{-(\alpha-\epsilon)d}$. Since by definition $P_{\text{error}}(\mathcal{E},\mathcal{D}^a,i,d)\leq P_{\text{anyerr}}(\mathcal{E},\mathcal{D}^a,i,d)$, this means that there is a uniform bound $K^{\mathcal{E}}$ as well. □

We immediately get the following corollary by applying Theorem 5.4.1 to Theorem 5.3.3

Corollary 5.4.1 *For a memoryless channel with block random coding exponent $E_r(R)$,*

$$C_{anytime}(E_r(R)\log_2 e)\geq R\log_2 e$$

Moreover, almost every deterministic code of a given rate achieves the corresponding α or better.

5.5 Discussion

In this chapter we have introduced a new parametric sense of reliable transmission and its associated capacity that we call anytime capacity. This capacity lies between zero-error capacity and Shannon classical capacity. It requires that for a given encoder, the decoder can make the error probability approach zero by waiting sufficiently long before declaring its estimate of a bit. Eventually, every bit is estimated correctly. We then showed that this capacity is related to error exponents and showed how a random coding argument could also be used to show the existence of deterministic codes satisfying our requirements. The work here raises many interesting unresolved issues.

As we have stated earlier, the definition we have given for anytime capacity is an operational one. Unlike Shannon capacity for which a simple characterization in terms of maximizing mutual information is available, we do not yet have any direct way of computing, even in principle, the anytime capacity of a general channel. Because of the connection to error exponents, for which optimal values are also not generally available, it is unclear whether a direct way of calculating the exact anytime capacity exists.

5.5.1 Better Estimates

One might be wondering why we used the block random coding exponent rather than the expurgated one [23] to get a lower bound on anytime capacity. The problem is that it is unclear how the expurgation procedure can be generalized to our infinite encoders. It is not possible to just throw "undesirable" branches of the tree away because asymptotically all branches are equally good! But even an asymptotically good branch can be initially bad and can be easily confused with another branch. Whether this is a fundamental difficulty with anytime codes that makes the α different than the best possible block error exponent remains an open problem.

This is particularly important in light of the tradeoff that can be made in practical systems regarding where to put error correction in a system. We could envision deploying anytime coding immediately after source coding and then count on the channel code to merely deliver a low probability of error in the Shannon sense. We could interpret the channel code's "reliable bitstream" as a binary symmetric channel with low crossover probability (or more accurately, in the spirit of concatenated coding [18]) and then expect the anytime code to eventually correct the rare errors that do occur. Or we could insist on deploying the anytime coding at the channel level itself. Given that expurgated and random coding bounds are the same in the noisier channel regime, this would seem to be a better strategy if better bounds for anytime capacity cannot be found for less noisy channels. At the moment, in the low-noise regime the anytime error rate for our random codes is dominated by the effect of randomness internal to the code rather than external noise.

5.5.2 Practical Codes

Because Lemma 5.2.1 tells us that we must be able to eventually get every bit correct, strict anytime codes must be "infinite in size" for any channel where anytime capacity differs from the classical zero-error capacity. The memory of the code must be infinite for it to be able to eventually correct errors made even in the distant past. If the memory were finite, then the delay could be upper bounded and we would be back to zero-error capacity.

Yet truly infinite objects do not exist in physical world. However, in real systems we may be satisfied with the error probability not tending all the way to zero, but rather to some extremely small number like 10^{-20} or so. At even a small $\alpha = \frac{1}{15}$, this requires a memory of approximately a thousand samples assuming a small K constant. A thousand samples might be too long to wait in the average case for a practical situation, but anytime decoding would presumably give very good estimates after a much shorter wait.

This issue of memory is not the only problem. The random code trees we have described are very large objects even if we truncate them to a moderate memory in terms of dependency on past samples. We would prefer to be able to have a compact description of the entire code. In some cases, we may be willing to live with a slower convergence of bit-error-probability to zero in exchange for a much more compact representation. It will be interesting to see if there are any "anytime" analogs of the computationally efficient new block coding and iterative decoding techniques used in Turbo Codes and Low Density Parity Check codes. Feedback may allow for significant computational savings in general just as it does in the binary erasure case.

5.5.3 Incremental Updates and Interconnections

An alternative interpretation of anytime decoding is an incremental one. Rather than giving out an estimate for the value of only those bits that had never been estimated before or giving updated estimates for all prior bits, the decoder can give updated estimates for just those bits which had previously been estimated incorrectly. These updates can be viewed as short variable length messages. Anytime reliability with a sufficiently fast decay on the probability of error implies that the expected number of bits that we will have to update is very small even as time goes on.

This suggests that it should be possible to think about interconnecting systems and placing some kind of intermediate transcoders between the noisy channels to recover from errors and boost signal reliability from end to end. We would like the capacity of such a serially interconnected system to be the capacity of the lowest-capacity component, with end-to-end delay behaving something like the sum over the channels. Such a theorem is needed to justify the "bit pipe" metaphor which is used for reliable transmission. We suspect that a theorem like that should be provable for independent channels. The parallel interconnection case is also interesting and in the longer term, we hope to be able to work towards a "calculus of channels" that allows us to better understand interconnections in general.

5.5.4 Splitting A Channel

If we are contemplating using many channels to transport a single bitstream from end-to-end, it is natural to also think of "splitting" a single channel and using it to transport many different bitstreams. With traditional Shannon or zero-error capacity there was not much that could be done beyond simple time-sharing. But, as Chapter 8 will show, the introduction of the reliability parameter in anytime capacity lets us pose a more interesting problem of transporting multiple bitstreams over a single channel. In general, we could imagine giving each of the bitstreams a different level of "error protection" in the sense of the rate at which the probability of error goes to zero with delay. This deserves a great deal of further study.

Chapter 6

An Information Transmission Theorem

In this chapter, we give a general result that covers unstable processes which tend to diverge exponentially. While we do not yet have as powerful a result as the traditional result separating source and channel coding given in Theorem 2.3.1, we have taken several steps in the right direction.

First, we will use our simple random walk example to illustrate how anytime capacity can allow us to track unstable processes across general noisy channels instead of just the seemingly special case of the erasure channel with feedback discussed in Section 2.4.3. Then we will state and prove a general result showing that a certain minimum anytime capacity at a certain reliability parameter is both sufficient and necessary for tracking unstable processes across noisy channels.

6.1 Tracking the Simple Random Walk over Noisy Channels

We will generate the bits $\{S_t\}$ using our source encoder F_t from (1.1). Now, suppose our channel has α-anytime-capacity larger than 1 for some α. Run our bits S_1^∞ through the rate 1 encoder to generate channel inputs. Now, to get estimates $\hat{X}_t$, we will use the anytime decoder directly as follows:

$$\begin{aligned} \hat{X}_t &= G_t(\mathcal{D}_t^a(B_1^t)) \\ &= \sum_{i=1}^{t}(2\left(\mathcal{D}_t^a(B_1^t)\right)_i - 1) \end{aligned}$$

To see that this does in fact track the source properly in the mean squared sense, we just compute:

$$\begin{aligned} E\left[(\hat{X}_t - X_t)^2\right] &= E\left[(\sum_{i=1}^{t}(2\left(\mathcal{D}_t^a(B_1^t)\right)_i - 2S_i))^2\right] \\ &\leq 4E\left[\left(\sum_{i=1}^{t}(\left(\mathcal{D}_t^a(B_1^t)\right)_i - S_i)^2\right)^2\right] \end{aligned}$$

$$= \quad 8E\left[\sum_{i=1}^{t}\sum_{j=1}^{i}(\left(\mathcal{D}_t^a(B_1^t)\right)_i - S_i)^2(\left(\mathcal{D}_t^a(B_1^t)\right)_j - S_j)^2\right]$$

We can denote the indicator for an error at bit i using the notation $\chi_i = ((\mathcal{D}_t^a(B_1^t))_i - S_i)^2$. This is 1 whenever the two are not equal and zero otherwise. Using this notation we have:

$$\begin{aligned}
E\left[(\hat{X}_t - X_t)^2\right] &\leq 8\sum_{i=1}^{t}\sum_{j=1}^{i} E[\chi_i\chi_j] \\
&= 8\sum_{i=1}^{t}\sum_{j=1}^{i} P(\chi_i = 1, \chi_j = 1) \\
&\leq 8\sum_{i=1}^{t}\sum_{j=1}^{i} \min(P(\chi_i = 1), P(\chi_j = 1)) \\
&\leq 8K\sum_{i=1}^{t}\sum_{j=1}^{i} \min(2^{-\alpha(t-i)}, 2^{-\alpha(t-j)}) \\
&< 8K\sum_{i=1}^{t}\sum_{j=1}^{i} 2^{-\alpha(t-j)} \\
&= 8K2^{-\alpha t}\sum_{i=1}^{t}\sum_{j=1}^{i} 2^{\alpha j} \\
&= 8K2^{-\alpha t}\sum_{i=1}^{t} \frac{2^{\alpha(i+1)} - 2^{\alpha}}{2^{\alpha} - 1} \\
&< 8K\frac{2^{-\alpha(t-1)}}{2^{\alpha} - 1}\sum_{i=1}^{t} 2^{\alpha i} \\
&= 8K\frac{2^{-\alpha(t-1)}}{2^{\alpha} - 1}\frac{(2^{\alpha(t+1)} - 2^{\alpha})}{2^{\alpha} - 1} \\
&< 8K\frac{2^{-\alpha(t-1)}2^{\alpha(t+1)}}{(2^{\alpha} - 1)^2} \\
&= 8K\frac{2^{2\alpha}}{(2^{\alpha} - 1)^2}
\end{aligned}$$

We never lose track of the simple random walk no matter how long we wait! As shown in the next sections, this basic argument can be generalized to cover more unstable sources, even those that are exponentially unstable as long as they do not grow too fast.

6.2 The Direct Part

We are now ready to introduce the major generalization of Theorem 3.2.1 to beyond the case of strong bit-streams. For weak bit-streams we have:

Theorem 6.2.1 *Consider a rate R stream of bits coming from a source code achieving finite* *expected distortion D_R with $\Delta^+(d)$. If there is a suitable decay function $f(d)$ satisfying:*

$$\lim_{\delta' \to \infty} \sum_{n=0}^{\infty} f(\delta' + \frac{n}{R})\Delta^+(\frac{n}{R}) = 0$$

Then the source code can be successfully transmitted across a noisy channel with finite expected distortion as long as the channel has $C_{anytime}(f) > R$. Furthermore, the end-to-end expected distortion can be made arbitrarily close to D_R if we are willing to tolerate enough additional end-to-end delay δ'.

Proof: Let δ be the delay of the original source code (F, G). Assume that we are willing to tolerate upto $\delta' \geq 0$ units of additional delay. Choose a rate R anytime channel code and feed the source bit stream into it. Recall that one of properties of anytime codes is that eventually, all the early bits are decoded correctly. So we will get our estimates of the source X_t at time $t + \delta + \delta'$ by using

$$\check{X}_t = \hat{x}_t(\left(\mathcal{D}^a_{\lfloor \frac{t+\delta+\delta'}{\tau} \rfloor}(B_1^{\lfloor \frac{t+\delta+\delta'}{\tau} \rfloor})\right)_1^{\lfloor (t+\delta)R \rfloor})$$

To bound the expected distortion, we just notice:

$$\begin{aligned}
E[\rho(X_t, \check{X}_t)] &\leq D_R + 0P(S_1^{\lfloor (t+\delta)R \rfloor} = \hat{S}_1^{\lfloor (t+\delta)R \rfloor}) \\
&\quad + \Delta^+(\frac{1}{R})P(S_{\lfloor (t+\delta)R \rfloor} \neq \hat{S}_{\lfloor (t+\delta)R \rfloor} | S_1^{\lfloor (t+\delta)R \rfloor - 1} = \hat{S}_1^{\lfloor (t+\delta)R \rfloor - 1}) \\
&\quad + \Delta^+(\frac{2}{R})P(S_{\lfloor (t+\delta)R \rfloor - 1} \neq \hat{S}_{\lfloor (t+\delta)R \rfloor - 1} | S_1^{\lfloor (t+\delta)R \rfloor - 2} = \hat{S}_1^{\lfloor (t+\delta)R \rfloor - 2}) \\
&\quad + \cdots + \Delta^+(t+\delta)P(S_1 \neq \hat{S}_1) \\
&\leq D_R + \sum_{n=0}^{\lfloor (t+\delta)R \rfloor} \Delta^+(\frac{n}{R})P(S_{\lfloor (t+\delta)R \rfloor - n} \neq \hat{S}_{\lfloor (t+\delta)R \rfloor - n}) \\
&< D_R + \sum_{n=0}^{\lfloor (t+\delta)R \rfloor} Kf(\delta' + \frac{n}{R})\Delta^+(\frac{n}{R}) \\
&\leq D_R + K\sum_{n=0}^{\infty} f(\delta' + \frac{n}{R})\Delta^+(\frac{n}{R})
\end{aligned}$$

By assumption on the decay function f, this is clearly finite and also tends to the source code's D_R as we let the additional delay δ' go to infinity. $\square$

Theorem 6.2.1 is in terms of general anytime capacity parametrized by a suitable decay function. It is clear that for $\Delta^+(d) \leq M 2^{-\alpha d}$, the function $f(d) = 2^{-(\alpha+\epsilon)d}$ is suitable as long as $\epsilon > 0$ since:

$$\begin{aligned}
&\lim_{\delta' \to \infty} \sum_{n=0}^{\infty} f(\delta' + \frac{n}{R})\Delta^+(\frac{n}{R}) \\
\leq\ & M \lim_{\delta' \to \infty} \sum_{n=0}^{\infty} 2^{-(\alpha+\epsilon)(\delta' + \frac{n}{R})} 2^{\alpha \frac{n}{R}} \\
=\ & M \lim_{\delta' \to \infty} 2^{-(\alpha+\epsilon)\delta'} \sum_{n=0}^{\infty} 2^{-\frac{\epsilon}{R}n}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{M}{1-2^{-\frac{\epsilon}{R}}} \lim_{\delta' \to \infty} 2^{-(\alpha+\epsilon)\delta'} \\
&= 0
\end{aligned}
$$

So by implicitly using the encoders from Theorem 4.1.1 and the growth rate property from Theorem 3.3.2, we immediately get the following corollary.

Corollary 6.2.1 *A scalar discrete-time unstable linear Markov process with parameter A driven by bounded noise can be tracked (with finite expected η-error distortion) across a noisy channel if there is an $\epsilon > 0$ for which $C_{anytime}(\eta \log_2 A + \epsilon) > \log_2 A$ for the channel. In particular, if $C_{anytime}(2 \log_2 A + \epsilon) > \log_2 A$, then we can track in the mean-squared sense.*

Since this clearly also holds in the case when $A = 1$, we have resolved Berger's comment about transporting the source code for the Wiener process across a noisy channel without a noiseless feedback link. Since Theorem 5.3.3 holds even for discrete memoryless channels without any feedback at all, we know that it is theoretically possible to get finite mean squared error as time goes to infinity for the Wiener process across a noisy link! This is accomplished by cascading Berger's source code with an anytime channel code for the noisy channel and handling the updates appropriately.

6.3 The Converse Part: Simple Case

It is unclear what the tightest and most general converse to Theorem 6.2.1 is. However, we can say quite a bit if we focus on the scalar unstable Markov processes of Corollary 6.2.1.

We begin by stating a converse of Corollary 6.2.1 for the bounded driving noise case.

Theorem 6.3.1 *Given a noisy channel, if there exists a joint source/channel code $(\mathcal{E}, \mathcal{D})$ which "tracks" all scalar discrete-time unstable linear Markov processes $\{X_t\}$ with parameter A driven by any bounded noise signal $-\frac{\Omega}{2} \leq W_t \leq \frac{\Omega}{2}$ so that:*

$$P(|\hat{X}_t - X_t| \geq \Delta) \leq f(\Delta)$$

then for all $\epsilon > 0$ the channel has $C_{anytime}(\tilde{f}) > \log_2 A - \epsilon$ where $\tilde{f} = f(\delta' 2^{d \log_2 A})$ for some constant $\delta' > 0$.

In particular, if the original process can be tracked in the expected η-distortion sense, then $\tilde{f}(d)$ can be made to be like $2^{-(\eta \log_2 A)d}$ giving us an α-anytime-capacity $C_{anytime}(\eta \log_2 A) > \log_2 A - \epsilon$.

Notice that we have stated the converse in its constructive contrapositive sense rather than as the traditional "only if" type of statement. We do this because our proof is constructive. We show how to take a joint source/channel encoder $\mathcal{E}$ and decoder $\mathcal{D}$ that tracks the source over the channel and use it to construct a channel encoder $\mathcal{E}'$ and anytime decoder $\mathcal{D}^a$ that has rate $\log_2 A - \epsilon$ and a probability of error that decays appropriately fast. To do this, we describe a way of encoding a stream of random bits S_1^∞ into something that looks like a scalar Markov process with parameter A driven by a bounded noise signal. The real issue will be to recover all these bits reliably from noisy estimates of this simulated Markov process. The full proof is involved and needs the use of some new representations of the real numbers.

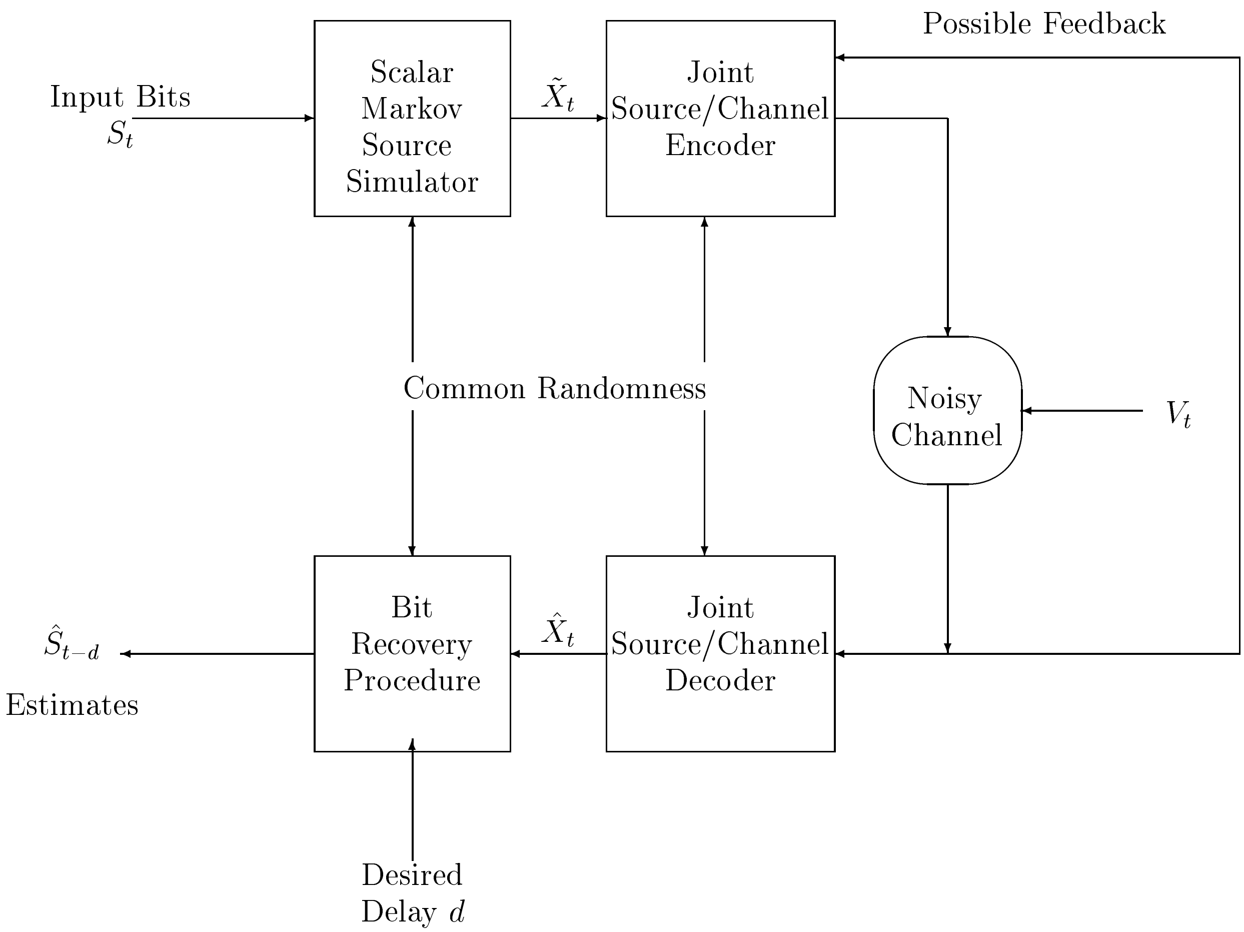

Figure 6-1: Using a source simulator to constructively prove the converse of the Information Transmission Theorem

One interesting feature of our proof for the converse statement expressed in Theorem 6.3.1 is that it does not use a per-letter or other characterization of the anytime capacity or the rate-distortion function for the source. This is in contrast to standard proofs of the traditional converse to the information transmission theorems which rely crucially on the data processing inequality and the fact that we have a mutual-information characterization of Shannon channel capacity.[66] Ours is instead an explicit construction that relates the two operational notions directly to each other.

6.3.1 The Encoder

We now construct the anytime encoder $\mathcal{E}'$ to use the input bits S_1^∞ to simulate the Markov source. For simplicity, we start with the case where $A = (2+\epsilon_1) = 2^{1+\epsilon_2}$ (so $\log_2 A = 1+\epsilon_2$) and aim for a rate 1 encoder. We use a "one bit" W_t signal that takes values $\pm\frac{\delta}{2}$ where δ is chosen to satisfy the bound on the noise. The mapping from an input bit to a realization of W_t in this case is obvious: $W_t(S_t) = \frac{\delta}{2}(2S_{t+1} - 1)$. We can use this to simulate the source:

$$\tilde{X}_{t+1} = A\tilde{X}_t + W_t \tag{6.1}$$

with initial condition $\tilde{X}_0 = 0$. We then feed this simulated source into the joint source/channel encoder $\mathcal{E}$ to get our channel encoder:

$$\mathcal{E}'_t(S_1^t) = \mathcal{E}_t(\tilde{X}_1^t) \tag{6.2}$$

where it is clear from equation (6.1) and our driving noise that the simulated source $\tilde{X}$ from 1 to t only depends on the first t input bits. Furthermore, we can write it out explicitly for $A = (2+\epsilon_1)$ as:

$$\tilde{X}_t = \sum_{i=1}^{t} A^{t-i} W_i = \frac{\delta}{2}\sum_{j=1}^{t}(2+\epsilon_1)^{t-j}(2S_j - 1)$$

For the case of general $A > 1$, we want to preserve the basic form of the source simulation. To generate the $\{W_t\}$ from the input bits, we think of W as a rate $R = \log_{2+\epsilon_1} A > 0$ encoder: formally a $(0,0)$ offset $(\frac{1}{\log_{2+\epsilon_1} A}, 1, \{0,1\}, \Re)$ memoryless transition map as follows:

$$\begin{aligned}
& W_t\left(S_{(\lfloor (t-1)\log_{2+\epsilon_1} A\rfloor + 1)}^{\lfloor t\log_{2+\epsilon_1} A\rfloor}\right) \\
= & \frac{\delta}{2}(2+\epsilon_1)^{(t\log_{2+\epsilon_1} A) - \lfloor t\log_{2+\epsilon_1} A\rfloor} \sum_{j=\lfloor (t-1)\log_{2+\epsilon_1} A\rfloor + 1}^{\lfloor t\log_{2+\epsilon_1} A\rfloor} (2+\epsilon_1)^{\lfloor t\log_{2+\epsilon_1} A\rfloor - j}(2S_j - 1)
\end{aligned}$$

Plugging the simulated noise in to get $\tilde{X}_t$ recursively gives us the following:

$$\begin{aligned}
\tilde{X}_t &= \sum_{i=1}^{t} A^{t-i} W_i \\
&= \sum_{i=1}^{t} A^{t-i}\frac{\delta}{2}(2+\epsilon_1)^{(i\log_{2+\epsilon_1} A) - \lfloor i\log_{2+\epsilon_1} A\rfloor} \sum_{j=\lfloor (i-1)\log_{2+\epsilon_1} A\rfloor + 1}^{\lfloor i\log_{2+\epsilon_1} A\rfloor} (2+\epsilon_1)^{\lfloor i\log_{2+\epsilon_1} A\rfloor - j}(2S_j - 1)
\end{aligned}$$

$$
\begin{aligned}
&= A^t \frac{\delta}{2} \sum_{i=1}^{t} (2+\epsilon_1)^{-\lfloor i \log_{2+\epsilon_1} A \rfloor} \sum_{j=\lfloor (i-1)\log_{2+\epsilon_1} A \rfloor + 1}^{\lfloor i \log_{2+\epsilon_1} A \rfloor} (2+\epsilon_1)^{\lfloor i \log_{2+\epsilon_1} A \rfloor - j} (2S_j - 1) \\
&= A^t \frac{\delta}{2} \sum_{i=1}^{t} \sum_{j=\lfloor (i-1)\log_{2+\epsilon_1} A \rfloor + 1}^{\lfloor i \log_{2+\epsilon_1} A \rfloor} (2+\epsilon_1)^{-j} (2S_j - 1) \\
&= A^t \frac{\delta}{2} \sum_{i=1}^{\lfloor t \log_{2+\epsilon_1} A \rfloor} (2+\epsilon_1)^{-i} (2S_i - 1) \\
&= \frac{A^t}{(2+\epsilon_1)^{\lfloor t \log_{2+\epsilon_1} A \rfloor}} \frac{\delta}{2} \sum_{i=1}^{\lfloor t \log_{2+\epsilon_1} A \rfloor} (2+\epsilon_1)^{\lfloor t \log_{2+\epsilon_1} A \rfloor - i} (2S_i - 1)
\end{aligned}
$$

resulting in:

$$
\tilde{X}_t = (2+\epsilon_1)^{t \log_{2+\epsilon_1} A - \lfloor t \log_{2+\epsilon_1} A \rfloor} \frac{\delta}{2} \sum_{i=1}^{\lfloor t \log_{2+\epsilon_1} A \rfloor} (2+\epsilon_1)^{\lfloor t \log_{2+\epsilon_1} A \rfloor - i} (2S_i - 1) \tag{6.3}
$$

To see that this simulated driving noise signal stays bounded, we notice:

$$
\begin{aligned}
W_t\left(S_{(\lfloor (t-1)\log_{2+\epsilon_1} A \rfloor + 1)}^{\lfloor t \log_{2+\epsilon_1} A \rfloor}\right) &\leq \frac{\delta}{2}(2+\epsilon_1)^1 \sum_{j=\lfloor (t-1)\log_{2+\epsilon_1} A \rfloor + 1}^{\lfloor t \log_{2+\epsilon_1} A \rfloor} (2+\epsilon_1)^{\lfloor t \log_{2+\epsilon_1} A \rfloor - j} (2S_j - 1) \\
&\leq \frac{\delta}{2}(2+\epsilon_1) \sum_{j=\lfloor (t-1)\log_{2+\epsilon_1} A \rfloor + 1}^{\lfloor t \log_{2+\epsilon_1} A \rfloor} (2+\epsilon_1)^{\lfloor t \log_{2+\epsilon_1} A \rfloor - j} \\
&\leq \frac{\delta}{2}(2+\epsilon_1) \sum_{j=1}^{\lceil \log_{2+\epsilon_1} A \rceil} (2+\epsilon_1)^{j} \\
&< \frac{\delta}{2}(2+\epsilon_1)^{2+\lceil \log_{2+\epsilon_1} A \rceil}
\end{aligned}
$$

By symmetry, the same bound applies on the other side so we have:

$$
-\frac{\delta}{2}(2+\epsilon_1)^{2+\lceil \log_{2+\epsilon_1} A \rceil} < W_t < \frac{\delta}{2}(2+\epsilon_1)^{2+\lceil \log_{2+\epsilon_1} A \rceil}
$$

which can be made to fit the bound $\frac{\Omega}{2}$ by choosing an appropriately small δ. Therefore, we know that the original decoder $\mathcal{D}$ succeeds in constructing a process $\{\hat{X}_t\}$ such that $P(|\hat{X}_t - \tilde{X}_t| \geq \Delta) \leq f(\Delta)$. The key now is to extract reconstructions $\hat{S}_1^{l(t)}$ from $\hat{X}_t$. To do this, we will need to take a little digression into the world of representing real numbers as strings and vice-versa.

6.3.2 Real Numbers As Strings: Analog to Digital Conversion

There are many different ways of looking at the real numbers.[17] Dedekind cuts (a strict subset $\mathcal{C}$ of the rationals such that if $x < y \in \mathcal{C}$ then $x \in \mathcal{C}$) are the traditional way of constructing them. Implicit in this construction is the view that a real number is specified by comparisons with the countable set of rationals. This is also a good way of looking at the

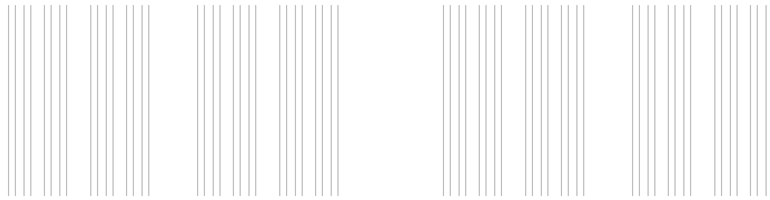

Figure 6-2: The Cantor set of possible binary strings when viewed as real numbers on the unit interval using $\epsilon_1 = 0.3$. $\epsilon_1 = 0$ would correspond to the entire interval

binary representation of a real number. Each 1 or 0 represents the outcome of a comparison with a known rational number.

In general, given a positive real number x with a known upper bound M, we can extract a binary string s_1^∞ by applying the following infinite procedure:

1. Start with the empty string. Let $n = 1$ and $x' = x$.
2. Compare x' with a threshold T_n. Set $s_n = 0$ if x' is strictly less than the threshold, otherwise set $s_n = 1$.
3. Adjust x' by an offset O_n by setting $x' = x' - s_n O_n$.
4. Increment n and Goto 2.

If we only care about the first N bits in the string, the infinite procedure can be stopped after $n = N$. In addition, if we multiply x by some known constant $\beta > 0$, then simply multiplying all the thresholds and offsets by the same β would give an identical sequence of output bits. If x was instead shifted additively by some constant γ, then simply shifting T_1 and O_1 by that same γ would give an identical sequence as well. Finally, it should be clear that the above procedure is completely determined by the choice of thresholds T_1^∞ and offsets O_1^∞. For example, to get the traditional binary representation, we can set $T_n = O_n = 2^{-n} 2^{\lceil \log_2 M \rceil}$. The traditional binary representation has the additional property of being able to distinguish any two real numbers. That is, if $x \neq \check{x}$ then $s_1^\infty \neq \check{s}_1^\infty$.

We do not need this property since we only want to distinguish between the subset of real numbers that we care about. Let us review the real numbers generated by the simple source simulated by our encoder for the case $A = (2 + \epsilon_1)$. At time t, we have $\tilde{X}_t = \frac{\delta}{2} \sum_{i=1}^t (2 + \epsilon_1)^{t-i}(2S_i - 1)$. This is not necessarily positive, so we can shift it by $\delta \sum_{i=1}^t (2 + \epsilon_1)^{t-i} = \delta \frac{(2+\epsilon_1)^t - 1}{(2+\epsilon_1)-1} = \delta \frac{(2+\epsilon_1)^t - 1}{1+\epsilon_1}$ to get $\delta \sum_{i=1}^t (2 + \epsilon_1)^{t-i} S_i$ which is always non-negative regardless of the exact values of the bits S_1^t. We can further normalize this by multiplying by $\frac{(2+\epsilon_1)^{-t}}{\delta}$ to give us $\tilde{X}'_t = \sum_{i=1}^t (2 + \epsilon_1)^{-i} S_i$ which is always in the interval $[0, 1]$. The general case of $A > 1$ just introduces another multiplication by a known constant in the the interval $[1, 2 + \epsilon_1]$ as (6.3) shows.

In fact, if we look at $\tilde{X}'_\infty$ and view it as a mapping $\tilde{x}'_\infty(s_1^\infty) = \sum_{i=1}^\infty (2+\epsilon_1)^{-i} s_i$ its range is a Cantor set $\mathcal{X}$. By this, we mean that $\mathcal{X}$ has zero Lebesgue measure and has "holes" which are open sets containing no points of $\mathcal{X}$. To see this, we will first need to review the natural ordering on semi-infinite strings:

Definition 6.3.1 *The* lexicographical comparison *between semi-infinite binary strings s_1^∞ and $\check{s}_1^\infty$ is defined recursively as follows.*

- *If $s_i = \check{s}_i$ for every $i \geq 1$, then the infinite strings s_1^∞ and $\check{s}_1^\infty$ are equal.*
- *If $s_1 = 1$ while $\check{s}_1 = 0$ then $s_1^\infty > \check{s}_1^\infty$ regardless of the later bits.*
- *If $s_1 = \check{s}_1$ then $s_1^\infty > \check{s}_1^\infty$ if and only if $s_2^\infty > \check{s}_2^\infty$.*

It is clear that the lexicographical comparison defines a natural total ordering on infinite binary strings. The following lemma shows that this ordering is compatible with the natural ordering on the real numbers under the mapping $\tilde{x}'_\infty$.

Lemma 6.3.1 *Assume $\epsilon_1 > 0$. The mapping $\tilde{x}'_\infty$ is strictly monotonic: $s_1^\infty > \check{s}_1^\infty$ if and only if $\tilde{x}'_\infty(s_1^\infty) > \tilde{x}'_\infty(\check{s}_1^\infty)$ as well. Moreover, if two sequences s_1^∞ and $\check{s}_1^\infty$ first differ in position i, then*

$$|\tilde{x}'_\infty(s_1^\infty) - \tilde{x}'_\infty(\check{s}_1^\infty)| \geq (2+\epsilon_1)^{-i}\frac{\epsilon_1}{1+\epsilon_1} > 0$$

Proof: Suppose that $\tilde{x}'_\infty(s_1^\infty) > \tilde{x}'_\infty(\check{s}_1^\infty)$. Then we can expand them out:

$$\begin{aligned}
\tilde{x}'_\infty(s_1^\infty) &> \tilde{x}'_\infty(\check{s}_1^\infty) \\
\sum_{i=1}^{\infty}(2+\epsilon_1)^{-i}s_i &> \sum_{i=1}^{\infty}(2+\epsilon_1)^{-i}\check{s}_i \\
\sum_{j=1}^{N}(2+\epsilon_1)^{-i_j}s_{i_j} &> \sum_{j=1}^{N}(2+\epsilon_1)^{-i_j}\check{s}_{i_j}
\end{aligned}$$

Where the final inequality comes from dropping all the shared terms in the sum from both sides and then choosing $N \leq \infty$ and arranging it so that $i_j < i_{j+1}$. Notice that this means for all $i < i_1$, we must have $s_i = \check{s}_i$. Because comparisons are not effected by multiplying both sides by a constant, we can multiply both sides by $(2+\epsilon_1)^{i_1-1}$ to get:

$$\sum_{j=1}^{\infty}(2+\epsilon_1)^{-j}s'_j > \sum_{j=1}^{\infty}(2+\epsilon_1)^{-j}\check{s}'_j$$

where we adjust notation so that $s'_j = s_{i_k}$ whenever $i_k = j$ and zero otherwise. (similarly for $\check{s}'_j$) Now notice that:

$$\begin{aligned}
\sum_{j=1}^{\infty}(2+\epsilon_1)^{-j}s'_j &= (2+\epsilon_1)^{-1}s'_1 + \sum_{j=2}^{\infty}(2+\epsilon_1)^{-j}s'_j \\
&\geq (2+\epsilon_1)^{-1}s'_1
\end{aligned}$$

and similarly for $\check{s}'$ while simultaneously

$$\begin{aligned}
\sum_{j=1}^{\infty}(2+\epsilon_1)^{-j}s'_j &= (2+\epsilon_1)^{-1}s'_1 + \sum_{j=2}^{\infty}(2+\epsilon_1)^{-j}s'_j \\
&\leq (2+\epsilon_1)^{-1}s'_1 + \sum_{j=2}^{\infty}(2+\epsilon_1)^{-j} \\
&= (2+\epsilon_1)^{-1}s'_1 + \frac{(2+\epsilon_1)^{-1}}{1+\epsilon_1}
\end{aligned}$$

and similarly for $\check{s}'$. Now, we have:

$$(2+\epsilon_1)^{-1}s'_1 + \frac{(2+\epsilon_1)^{-1}}{1+\epsilon_1} \geq \sum_{j=1}^{\infty}(2+\epsilon_1)^{-j}s'_j > \sum_{j=1}^{\infty}(2+\epsilon_1)^{-j}\check{s}'_j \geq (2+\epsilon_1)^{-1}\check{s}'_1$$

By rearranging terms in the above inequality and multiplying both sides by a further $(2+\epsilon_1)^{-1}$ we have:

$$s'_1 - \check{s}'_1 + \frac{1}{1+\epsilon_1} > 0$$

Because $\epsilon_1 > 0$, we know $\frac{1}{1+\epsilon_1} < 1$. Since we have $s'_1 \neq \check{s}'_1$, we can conclude that $s_{i_1} = s'_1 = 1$ while $\check{s}_{i_1} = \check{s}'_1 = 0$. Therefore, by our definition of lexicographical ordering, $s_1^\infty > \check{s}_1^\infty$.

We can use a very similar argument to establish our bound on the differences while also establishing the other direction. Suppose that s_1^∞ and $\check{s}_1^\infty$ first differ in position i. Without loss of generality, assume that $s_i = 1$ while $\check{s}_i = 0$. Then

$$\begin{aligned}
\tilde{x}'_\infty(s_1^\infty) - \tilde{x}'_\infty(\check{s}_1^\infty) &= \sum_{j=1}^{\infty}(2+\epsilon_1)^{-j}s_j - \sum_{j=1}^{\infty}(2+\epsilon_1)^{-j}\check{s}_j \\
&= (2+\epsilon_1)^{-i} + \sum_{j=i+1}^{\infty}(2+\epsilon_1)^{-j}(s_j - \check{s}_j) \\
&\geq (2+\epsilon_1)^{-i} - \sum_{j=i+1}^{\infty}(2+\epsilon_1)^{-j} \\
&= (2+\epsilon_1)^{-i} - \frac{(2+\epsilon_1)^{-i}}{1+\epsilon_1} \\
&= (2+\epsilon_1)^{-i}\frac{\epsilon_1}{1+\epsilon_1} > 0
\end{aligned}$$

This proves the Lemma. □

The proof of Lemma 6.3.1 actually tells us even more than the fact that the mapping is monotonic. It mathematically shows the Cantor set gaps we illustrate in Figure 6-2. Since such positive measure gaps are between every pair of elements in $\mathcal{X}$, we know that $\mathcal{X}$ can contain no open sets. It must have Lebesgue measure zero and moreover, the mapping $\tilde{x}'_\infty$ is nowhere continuous if reinterpreted as a mapping from the unit interval (viewed under the normal binary representation of real numbers) to itself.

These gaps are particularly useful to us in extracting the bits s_1^∞ from a real number in $\mathcal{X}$. They let us choose thresholds T_1^∞ and offsets O_1^∞ which are not equal to each other.

Lemma 6.3.2 *If $x = \sum_{i=1}^{\infty}(2+\epsilon_1)^{-i}s_i$ where $\epsilon_1 > 0$ and $s_i \in \{0,1\}$, then we can use the procedure with $O_n = (2+\epsilon_1)^{-n}$ and any set of thresholds as long as $(2+\epsilon_1)^{-n} \geq T_n > (2+\epsilon_1)^{-n}(1 - \frac{\epsilon_1}{1+\epsilon_1})$ for every $n \geq 1$ in order to recover the bits exactly.*

Proof: We will proceed by induction. Consider the base case of s_1. If $s_1 = 1$ we know that $x \geq (2+\epsilon_1)^{-1} \geq T_1$ and hence we will recover the first bit. If $s_1 = 0$, then we know by Lemma 6.3.1 that $(2+\epsilon_1)^{-1} - x \geq (2+\epsilon_1)^{-1}\frac{\epsilon_1}{1+\epsilon_1}$ and hence $x \leq (2+\epsilon_1)^{-1} - (2+\epsilon_1)^{-1}\frac{\epsilon_1}{1+\epsilon_1} < T_1$. Once again, we recover the first bit.

Now, assume that we recover correctly all the bits from 1 to $i-1$. By the procedure, we know that:

$$\begin{aligned} x' &= x - \sum_{j=1}^{i-1} s_j O_j \\ &= \sum_{j=1}^{\infty} (2+\epsilon_1)^{-j} s_j - \sum_{j=1}^{i-1} (2+\epsilon_1)^{-j} s_j \\ &= \sum_{j=i}^{\infty} (2+\epsilon_1)^{-j} s_j \\ &= (2+\epsilon_1)^{-(i-1)} \sum_{j=1}^{\infty} (2+\epsilon_1)^{-j} s_{i+j-1} \end{aligned}$$

We can scale x' and the threshold T_i by $(2+\epsilon_1)^{i-1}$ which would give us:

$$x' = \sum_{j=1}^{\infty} (2+\epsilon_1)^{-j} s_{i+j-1}$$

and

$$\begin{aligned} (2+\epsilon_1)^{-1} &= (2+\epsilon_1)^{i-1}(2+\epsilon_1)^{-i} \\ &\geq (2+\epsilon_1)^{i-1} T_i \\ &> (2+\epsilon_1)^{i-1}(2+\epsilon_1)^{-i}(1 - \frac{\epsilon_1}{1+\epsilon_1}) = (2+\epsilon_1)^{-1}(1 - \frac{\epsilon_1}{1+\epsilon_1}) \end{aligned}$$

This is exactly like the base case of $i = 1$ and therefore, we can extract the bit s_i correctly as well. Thus, the Lemma is proved by induction. □

We would like our thresholds to be as "robust" as possible to noise and so will choose thresholds in the center of their acceptable range:

$$\begin{aligned} T_n &= (2+\epsilon_1)^{-n} - \frac{1}{2}(2+\epsilon_1)^{-n}\frac{\epsilon_1}{1+\epsilon_1} = (2+\epsilon_1)^{-n}\frac{2+\epsilon_1}{2+2\epsilon_1} \\ O_n &= (2+\epsilon_1)^{-n} \end{aligned}$$

The structure of $\mathcal{X}$ given in Lemma 6.3.2 tells us that that this scheme will work for getting the bits out of $\tilde{X}'_t$ for any value of t. Suppose that instead of starting the procedure with $x' = \tilde{X}'_t$ we start it with a noisy version $x' = \tilde{X}'_t + \zeta$. We know from the proof of Lemma 6.3.2 that we are guaranteed to get the same answers for the first l bits as long as $|\zeta| < \frac{1}{2}(2+\epsilon_1)^{-l}\frac{\epsilon_1}{1+\epsilon_1}$.

This shows that our mapping $\tilde{x}'_\infty$ encodes bits in such a way that they are protected against noise. However, this protection is unequal as the earlier bits get exponentially more protection than the later ones. This mapping's "robustness" makes it dramatically different from the mapping representing traditional binary encodings whose continuity results in almost no protection at all.

6.3.3 The Decoder

With that digression completed, we are ready to construct our anytime decoder $\mathcal{D}^a$ out of the original decoder $\mathcal{D}$. To extract estimates for the first $\lfloor t \log_{2+\epsilon_1} A \rfloor$ bits out of $\hat{X}_t$, simply apply the following procedure:

1. Let $n = 1$ and set $x' = \frac{1}{\delta A^t}\hat{X}_t + \frac{1+(2+\epsilon_1)^{-\lfloor t \log_{2+\epsilon_1} A \rfloor}}{2+2\epsilon_1}$.
2. Apply the decoding procedure of Lemma 6.3.2 using the thresholds $T_n = (2+\epsilon_1)^{-n}\frac{2+\epsilon_1}{2+2\epsilon_1}$ and offsets $O_n = (2+\epsilon_1)^{-n}$ to get out the first $\lfloor t \log_{2+\epsilon_1} A \rfloor$ bits.

Once again, let $\tilde{X}'_t = \frac{1}{\delta A^t}\tilde{X}_t + \frac{1+(2+\epsilon_1)^{-\lfloor t \log_{2+\epsilon_1} A \rfloor}}{2+2\epsilon_1}$ be the rescaled and shifted version of the true signal from the simulated source. If $x' = \tilde{X}'_t + \zeta$, we know from the previous section that the first i bits are guaranteed to be correct whenever $|\zeta| < \frac{\epsilon_1}{2+2\epsilon_1}(2+\epsilon_1)^{-i}$. i bits are received by time $\lceil \frac{i}{\log_{2+\epsilon_1} A} \rceil$ and so we have:

$$\begin{aligned}
P_{\text{error}}(\mathcal{E}', \mathcal{D}^a, d, i) &\leq P\left(|\zeta| > \frac{\epsilon_1}{2+2\epsilon_1}(2+\epsilon_1)^{-i}\right) \\
&= P\left(|\delta A^{\lceil \frac{i}{\log_{2+\epsilon_1} A} \rceil + d}\zeta| > \frac{\epsilon_1}{2+2\epsilon_1}(2+\epsilon_1)^{-i}\delta A^{\lceil \frac{i}{\log_{2+\epsilon_1} A} \rceil + d}\right) \\
&= P\left(|X_t - \hat{X}_t| > \frac{\delta\epsilon_1}{2+2\epsilon_1}(2+\epsilon_1)^{-i}(2+\epsilon_1)^{(\log_{2+\epsilon_1} A)(\lceil \frac{i}{\log_{2+\epsilon_1} A} \rceil + d)}\right) \\
&\leq P\left(|X_t - \hat{X}_t| > \frac{\delta\epsilon_1}{2+2\epsilon_1}(2+\epsilon_1)^{(\log_{2+\epsilon_1} A)d}\right) \\
&\leq f\left(\frac{\delta\epsilon_1}{2+2\epsilon_1}(2+\epsilon_1)^{(\log_{2+\epsilon_1} A)d}\right) \\
&= f\left(\frac{\delta\epsilon_1}{2+2\epsilon_1}2^{d\log_2 A}\right)
\end{aligned}$$

This establishes the main result since for all i we have the proper decay with constant $\delta' = \frac{\delta\epsilon_1}{2+2\epsilon_1} > 0$. By choosing ϵ_1 appropriately we have proved the first part of the theorem as stated.

The particular case of η-distortion is a simple consequence of Markov's inequality since:

$$P(|\hat{X}_t - X_t| \geq \Delta) \leq \frac{E[|\hat{X}_t - X_t|^\eta]}{\Delta^\eta}$$

and hence

$$\begin{aligned}
P_{\text{error}}(\mathcal{E}', \mathcal{D}^a, d, i) &\leq \frac{E[|\hat{X}_t - X_t|^\eta]}{(\delta' 2^{d\log_2 A})^\eta} \\
&= \frac{E[|\hat{X}_t - X_t|^\eta]}{\delta'^\eta} 2^{-(\eta \log_2 A)d}
\end{aligned}$$

establishing the result. □

6.4 The Converse: More General Driving Noise Distributions

In proving Theorem 6.3.1, we assumed that our simple two-point support simulated noise was an acceptable realization for W_t. This is fine if all the joint source/channel encoder requires is that the noise be bounded. However, if we allow ourselves the use of common randomness, we can deal with more general situations as well. To do this, we use the ideas discussed in Appendix C just as we did in showing how the rate-distortion bound is achievable in the limit of large delays in Section 4.2.

6.4.1 "Robust" Joint Source/Channel Encoders

The joint source/channel code assumed in Theorem 6.3.1 works for any bounded input $\{W_t\}$ and can be thought of as a code that is very robust to the distribution. We can also consider codes that are less robust. In particular, codes which work as long as W_t has a distribution that is within an l_1 ball of a nominal distribution with a well-defined continuous density.

Extending Theorem 6.3.1 to this case involves approximating the density for W_t with a piecewise constant density very closely (to within the tolerance of the joint source/channel encoder) using Theorem C.2.1. By Examples C.1.1 and C.1.2, we know that we can additively construct such a piecewise constant density out of the pairs of δ functions used to encode our bits in the proof of Theorem 6.3.1 together with some common randomness. The resulting $\{W_t\}$ can be fed into the linear system given in (6.1) and the result sent into the joint source/channel encoder. By Lemma C.2.3 and common randomness, we can subtractively isolate the contributions to $\{\tilde{X}_t\}$ that come from the encoded bits. Therefore, if the joint source/channel decoder gives us estimates to some distance from the $\{\tilde{X}_t\}$ process, we can extract from them estimates that are the same distance away from the $\{\tilde{X}_t\}$ that would have arisen in Theorem 6.3.1. Thus, we get the following corollary:

Corollary 6.4.1 *Given a noisy channel, if there exists a robust joint source/channel code $(\mathcal{E}, \mathcal{D})$ that tracks all scalar discrete-time unstable linear Markov processes with parameter A driven by i.i.d. noise signals $\{W_t\}$ whose distributions are within a Ω ball in l_1 from a nominal distribution with Riemann integrable density so that:*

$$P(|\hat{X}_t - X_t| \geq \Delta) \leq f(\Delta)$$

then for all $\epsilon > 0$ the channel has $C_{anytime}(\tilde{f}) > \log_2 A - \epsilon$ if common randomness is present. Here $\tilde{f} = f(\delta' 2^{d \log_2 A})$ for some constant $\delta' > 0$ which in general depends on Ω as well as ϵ.

6.4.2 No Robustness: Exact Simulation

In the case where the joint source/channel code is only guaranteed to work for the nominal distribution, we can still do the approximation procedure of the previous section, but occasionally will have to not use the input bit and will instead just use the residual random variable that we get from the construction of Theorem C.2.1. This rare event (which is known to both the channel encoder and decoder) is interpreted as a simulated erasure channel with feedback. The input bits go into a simulated queue at rate R and are then used at rate $R' > R$ with common randomness to generate the samples of $\{W_t\}$ that are drawn from the nominal density. The common randomness insures that the anytime encoder and decoder stay synchronized with each other and can extract all the bits sent.

It is clear that this simulated erasure channel with feedback is going to cost us something in the rate at which the probability of error is going to zero. Notice that an error will occur if either there is a problem reconstructing from the system of Corollary 6.4.1, or if the bit in question was held up in the simulated input queue. Let us focus on the queue part. We choose $(\frac{R'}{R} - 1) = Ue$ where U is a large number and e is the probability of simulated erasure. We choose U and e so that Ue is still small enough so that the rates R' and R are close in absolute terms.

$$\begin{aligned}
& P(\text{bit from } i\tau \text{ units ago still in queue}) \\
= \; & P(\text{QueueLength} \geq i) \\
\leq \; & \sum_{j=1}^{\infty} P(\sum_{k=1}^{j} B_k R \geq jR' + i) \\
= \; & \sum_{j=1}^{\infty} P(\sum_{k=1}^{j} B_k \geq j\frac{R'}{R} + \frac{i}{R})
\end{aligned}$$

We can bound this probability using the Chernoff bound like we did in section 4.5.2. Doing that, we get:

$$\begin{aligned}
s &= \ln\left(\frac{\frac{R'}{R}(\frac{1}{e} - 1)}{\frac{R'}{R} - 1}\right) \\
&= \ln\left(\frac{(1 + Ue)(\frac{1}{e} - 1)}{Ue}\right) \\
&= \ln(\frac{1}{e} - 1) + \ln(1 + Ue) - \ln(Ue) \\
&= \ln(\frac{1}{e} - 1) + \ln(\frac{1}{Ue}) + \ln(1 + Ue)
\end{aligned}$$

which can be made arbitrarily large given that Ue is still small. So we call it a large multiple U' of the rate R and plug it back in:

$$\begin{aligned}
& P(\text{bit from } i\tau \text{ units ago still in queue}) \\
\leq \; & \sum_{j=1}^{\infty} \kappa^j e^{-s\frac{i}{R}} \\
< \; & e^{-U'i}
\end{aligned}$$

where U' is as large as we want to make it based on our choice of e and U. The probability of the bit i not being ready dies exponentially as fast as we want it to as long as we make e small enough. If a bit is delayed by $i\tau$ units in the queue, it is as though it was received $i\tau$ units later. And so the probability of error from the system of Corollary 6.4.1 would be as $f(\delta' 2^{(d-i\tau)\log_2 A})$. Putting it all together we have that the probability of error is bounded above by:

$$\sum_{i=0}^{\infty} 2^{-U'i} f(\delta' 2^{(d-i\tau)\log_2 A})$$

giving us:

Corollary 6.4.2 *Given a noisy channel, if there exists a joint source/channel code* $(\mathcal{E}, \mathcal{D})$ *which tracks a scalar discrete-time unstable linear Markov processes with parameter* A *driven by i.i.d. noise signals* $\{W_t\}$ *whose distribution has a density is Riemann integrable so that:*

$$P(|\hat{X}_t - X_t| \geq \Delta) \leq f(\Delta)$$

then for all $\epsilon > 0$ the channel has $C_{anytime}(\tilde{f}) > \log_2 A - \epsilon$ if common randomness is allowed to be used by the encoder and decoder. Here $\tilde{f} = \sum_{i=0}^{\infty} 2^{-U'i} f(\delta' 2^{(d-i\tau)\log_2 A})$ for all constants $U' > 0$ and some constant $\delta' > 0$ which in general depends on both U' and Ω as well as ϵ.

If we restrict our attention in Corollary 6.4.2 to α-anytime capacity where all we want is an exponential decay, then we know that by choosing U' large enough, the exponential coming from α will dominate the probability of error.

6.5 Converse For Performance: Not There Yet

Our new converse information transmission theorem 6.3.1 together with the extensions given by Corollaries 6.4.1 and 6.4.2 just relate the anytime capacity of the channel at a given α to whether or not we can track with finite expected distortion. Unlike the standard converse in Theorem 2.3.1, our theorems do not mention the actual performance achieved by the code. We might initially conjecture:

Conjecture 6.5.1 *For a channel, independent unstable Markov source $\{X_t\}$ with parameter A, and η-distortion measure, the performance of any pair of joint source-channel encoder/decoder pairs $\mathcal{E}, \mathcal{D}$ is bounded as follows: $E[\rho(X, \hat{X})] \geq R^{-1}(C_{anytime}(\eta \log_2 A))$ where $R^{-1}(C)$ is the distortion-rate function (the inverse of the rate-distortion function) which we evaluate at the appropriate anytime capacity of the channel. Moreover, we can get arbitrarily close to $R^{-1}(C)$ by using a good source coder followed by a good anytime channel code.*

Given our work on source coding in Chapter 4, it should be clear that we can achieve the direct part of this theorem by using an appropriate source code and Theorem 6.2.1. The converse side is unfortunately false as the following counterexample shows.

6.5.1 Counterexample

The idea of the counterexample is to consider a noisy channel that has two independent parts operating in parallel. The first is a regular binary erasure channel with a small erasure probability. The other is a real erasure channel.

Real-valued erasure channel

Definition 6.5.1 *The* real erasure channel *with erasure probability e is a memoryless noisy channel with $\mathcal{A} = \Re$, $\mathcal{B} = \Re \cup \{\emptyset\}$ and $P(a = b) = (1 - e)$ and $P(b = \emptyset) = e$.*

Since we know that we can embed an infinite number of binary digits into a single real number using the constructions of Section 6.3.2, it is easy to see that the regular Shannon capacity of this channel is ∞. Moreover, it is clear that regardless of the input rate, the optimal anytime encoding strategy is to encode all the bits received so far into each channel input. Upon reception of even a single non-erased symbol, all the past bits are known exactly at the anytime decoder. And so we know that we can only make an error in the case that we have a string of erasures between the time the bit was received and the current time. Thus for every bit-rate, the probability of error decays no faster than $e^{\frac{d}{\tau}}$ giving us:

Theorem 6.5.1 *For a real erasure channel with inter-use time τ and erasure probability $0 < e < 1$, we have:*

$$C_{anytime}(\alpha) = \begin{cases} \infty & \textit{if } \alpha \leq -\frac{\log_2 e}{\tau} \\ 0 & \textit{otherwise} \end{cases}$$

Combined channel

If the erasures on the regular binary erasure sub-channel happen less frequently than the erasures on the real-valued erasure sub-channel, then the anytime capacity looks like:

$$C_{\text{anytime}}(\alpha) = \begin{cases} \infty & \text{if } \alpha \leq -\frac{\log_2 e_{\Re}}{\tau} \\ f(\alpha) < \infty & \text{if } -\frac{\log_2 e_{\Re}}{\tau} < \alpha < -\frac{\log_2 e_b}{\tau} \\ 0 & \text{otherwise} \end{cases}$$

where $e_{\Re}$ is the probability of erasure on the real-sub channel and e_b on the binary one.

The important thing in this combined channel is that there is a region of α for which the anytime capacity is finite.

Performance

Now, imagine that we have an η large enough and $A > 1$ close enough to 1 so that:

$$-\frac{\log_2 e_{\Re}}{\tau} < \eta \log_2 A < -\frac{\log_2 e_b}{\tau}$$

while for the simple binary erasure channel with erasure probability e_b considered in isolation:

$$C_{\text{anytime}}(\eta \log_2 A) > \log_2 A$$

Then we know that there exists a coding system which just uses the binary part of this channel which achieves finite expected η-distortion, even if we use a causal code. Moreover, there is no way to achieve finite expected η-distortion if we restricted ourselves to the real part of the channel, infinite though the rate may be!

However, by using both parts of the channel, we can easily achieve an expected η-distortion that tends to 0 as we let ourselves use larger and larger delays. To do this, we use any causal code we want that achieves a finite expected η-distortion K using only the binary erasure part. Then, on the real-valued part, we send an infinite precision quantization of every sample in X_1^t. This is done by encoding every bit of all past X values into a single real number using the paradoxical properties of a "Code Infinity" construction analogous to that described in Section 2.2.1. If we are willing to wait for d time units, we know:

$$\begin{aligned} & \lim_{d\to\infty} E[|X_t - \hat{X}_t|^\eta] \\ \leq\ & \lim_{d\to\infty} (0P(\text{not all erasures from } d \text{ units ago}) + KP(\text{all erasures from } d \text{ units ago})) \\ \leq\ & \lim_{d\to\infty} \left(0(1 - e_{\Re}^{\frac{d}{\tau}}) + K e_{\Re}^{\frac{d}{\tau}}\right) \\ =\ & \lim_{d\to\infty} K e_{\Re}^{\frac{d}{\tau}} \\ =\ & 0 \end{aligned}$$

This happens despite the fact that the anytime capacity of the combined system at $\alpha = \eta \log_2 A$ is finite. Since the rate-distortion function is infinite for this real-valued process at $D = 0$, we have disproved the conjecture.

6.5.2 What We Do Know About Performance

At the moment, the only converse we have for infinite horizon performance is the appropriate version of the regular Shannon converse which says that we can do no better than the rate-distortion function evaluated at the classical Shannon capacity for the channel. If we had a joint source/channel code which violated this bound in the infinite horizon performance for some specified end-to-end delay, it would also violate the bound if we just considered a long finite horizon truncation of it and increased the rate by the infinitesimal factor required to compensate for the delay at the very end of the finite horizon piece. This would then violate the appropriate one-shot bound for this long finite-horizon segment and create a contradiction.

Notice that in the counterexample presented above we achieve that limit since the Shannon capacity of the combined channel is infinite. It is unclear whether the combination of our anytime converse and the regular Shannon converse is tight.

6.6 Discussion

In this chapter, we have presented the keystone result of this thesis. We have shown that our new notion of anytime capacity is both necessary and sufficient in order to communicate delay-sensitive bitstreams emerging from unstable Markov sources. In particular, we have established that such streams require not just a channel with sufficient bit-rate, but a high enough α parameter as well. Furthermore, the converse was given constructively and relates operational notions directly to each other without having to pass through a formal characterization in terms of mutual information or any other quantity. Yet once again, this result has raised many interesting issues. A few will be elucidated in the next two chapters, but some still need more study.

6.6.1 "Recursive" Formulations of The Direct Part

The constructions given so far for how to cascade the anytime-decoder with the source decoder are not recursive at all even though the source decoder often is. Instead, we have the source decoder recomputing from scratch using the most current estimates of all the bits sent. But we know that asymptotically the estimate for most bits are not going to change at all and Section 5.5.3 tells us that we can think of the anytime decoder as producing small finite sized update messages.

It is clear that the direct part can interpreted in a recursive way where the source decoder is made able to use corrections on past information to update its current estimates rather than having to always recompute everything from scratch. This should be explored more fully in future work.

6.6.2 A Path To A Tighter Converse

The story of the separation theorem will not be complete until we have a tight converse. An approach which might succeed is to first conceptually "split" the channel and ask it to transport two bitstreams. The first should have just the required α and rate to track the source with finite, but possibly large, distortion. The second bitstream should be allocated the maximal rate "leftover" after providing for the first one. Then, we should study the "successive refinement" [16, 53] of the source process into two streams matching those rates. By optimizing over the split and allowing for sufficient delay, it might be possible to get a tight converse.

Also, we need to formulate converses for more general classes of delay-sensitive processes than just the scalar Markov case. The linear vector case should be straightforward, but more general ones need much further thought.

Chapter 7

Control and Feedback

In this chapter, we look at the case of channels with noiseless feedback. In the first section, we make the connection with controlling unstable linear systems and show how that lets us extend our information transmission theorem 6.3.1 to control situations over channels with noiseless feedback. In some cases, the control situation has clearly optimal solutions and so allows us to get good bounds on the anytime capacity for channels where the encoders are allowed access to noiseless feedback. In the second half of this chapter, we evaluate the anytime capacity for both binary erasure and AWGN channels with noiseless feedback.

7.1 Feedback as Control

The notion of feedback is central to control theory and in fact, much of the work in this thesis is implicitly motivated by ideas from control. Here, we make the connection explicit by illustrating the relationship between the problem of controlling an unstable linear system over a noisy channel[63] depicted in Figure 7-1 and our standard problem of estimating an unstable process over a noisy channel depicted in Figure 7-2. Notice that in both cases, we allow for the encoder (or observer) side to have access to noiseless feedback from the channel, delayed by 1 channel step of time to avoid any causality problems.

7.1.1 Control With A Communication Constraint

To talk about the control problem, we first need a slight modification of Definition 2.1.1:

Definition 7.1.1 *Given a real number A, and functions W_t from $[0,1]$ to $\Re$, the* scalar discrete-time controlled Markov source with parameter A and noise W *is defined by:*

$$\begin{aligned}\tilde{X}_0 &= 0\\ \tilde{X}_t &= A\tilde{X}_{t-1} + W_t + U_{t-1}\end{aligned}$$

This can be expressed without recursion as:

$$\tilde{X}_t = \sum_{i=1}^{t} A^{t-i}(W_i + U_{i-1})$$

The general objective is to keep $|\tilde{X}_t|$ "suitably small" by design of the total system. The details of what is meant by "suitably small" are discussed in Section 7.1.2. The control

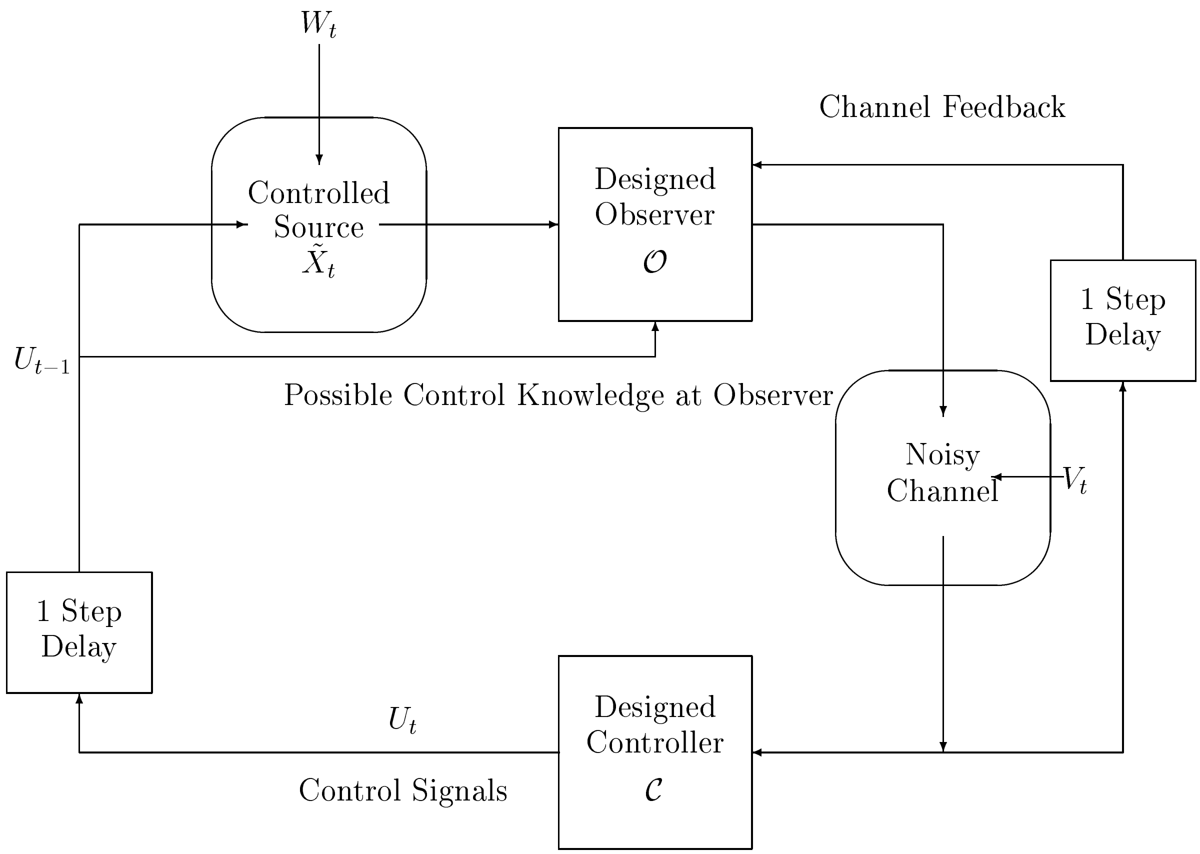

Figure 7-1: The control problem over a noisy channel

signals $\{U_t\}$ are determined "in closed loop" as shown in Figure 7-1 and are thus constrained to be a function only of the output stream of the communication channel.

Notice that there are actually three different potential feedback loops in Figure 7-1:

1. The tight loop from observer $\mathcal{O}$ through the channel and then right back to the observer with a one step delay.

2. The medium loop from observer $\mathcal{O}$ through the channel and the controller $\mathcal{C}$ coming back to the observer with a one step delay.

3. The big loop from observer $\mathcal{O}$ through the channel and the controller $\mathcal{C}$, continuing on with a one step delay through the controlled system and finally coming back to the observer.

Only the third loop is essential to the control problem. The first two are related to the concept of "equimemory"[22] and their implications are discussed more extensively in [63]. Notice that the feedback information from the first loop (the channel output signal) is actually sufficient to calculate the control signals that would be received back through the second loop since both the observer and controller are designed systems and hence presumably known.

The key observation is that knowledge of the system input signals and the system output is sufficient to calculate the 0-input response of the system since we assume that the controlled system is known exactly. Yet the 0-input response of the system is always a realization of an uncontrolled Markov process and vice versa. This is reminiscent of Witsenhausen's discussion of the separation between control and estimation in control problems with classical information patterns [70]. In the next section we establish the equivalence of the control problem of Figure 7-1 to the estimation problem in Figure 7-2.

7.1.2 What Stability Means

Keeping $|\tilde{X}_t|$ "suitably small" when the underlying controlled system is unstable is referred to as stabilizing the system. If the entire system (including all components we design) is deterministic and known to the designer in advance, then it can be possible to drive the state $\tilde{X}$ all the way to zero. However, if there is any persisting excitation in the system, achieving such a strict sense of stability is impossible. So, here is a looser definition:

Definition 7.1.2 *We call the complete system* boundedly stable *if there exists a constant K so that $|\tilde{X}_t| \leq K$ for all possible realizations of the uncertainty.*

This is often possible if all the uncertain signals are bounded, but if there is true randomness involved then it is hard to get these sort of rigid guarantees. In cases with randomness, a yet looser family of definitions is often appropriate:

Definition 7.1.3 *We call the complete system* η-stable *($\eta > 0$) if there exists a constant K so that $E[|\tilde{X}_t|^\eta] \leq K$ for all possible realizations of the nonprobabilistic uncertainty, with the expectation taken over all the probabilistic uncertainty. The case of $\eta = 2$ is also called* mean square stability*.*

In the case when all the primitive variables are functions of a common probability space, we can in principle evaluate η-stability using the complementary distribution function $P(|\tilde{X}_t| > x)$. The larger η is, the faster the complementary distribution function needs to go to zero as a function of x. However, we are also interested in cases where some of the primitive variables (namely the $\{W_t\}$) do not necessarily come from a probability distribution but are known to be bounded. To deal with this sort of mixed situation, where the system driving noise is arbitrary whereas the channel noise is stochastic, we think of the complementary distribution function for $|\tilde{X}_t|$ as being indexed by the realization of $\{W_t\}$.

Alternatively, we can think of the $\tilde{X}_t \in \Upsilon_t$ where Υ_t is a set valued random variable that depends only on the randomness in the system. The value for $\tilde{X}_t$ depends on the actual realization of the arbitrary driving noise and can be anywhere within Υ_t. As a result, we can be conservative and let

$$\Lambda_t = \sup_{\tilde{X}_t \in \Upsilon_t} |\tilde{X}_t| \tag{7.1}$$

be the random variable representing the upper bound on $|\tilde{X}_t|$. With this definition, it is clear that having $E[\Lambda_t^\eta] \leq K < \infty$ implies η-stability. As such, it is often the complementary distribution function of Λ_t that we are interested in. In particular, we can write $P(|\tilde{X}_t| > x) \leq f_t(x)$ whenever $P(\Lambda_t > x) \leq f_t(x)$ without having to specify the exact realization of $\{W_t\}$.

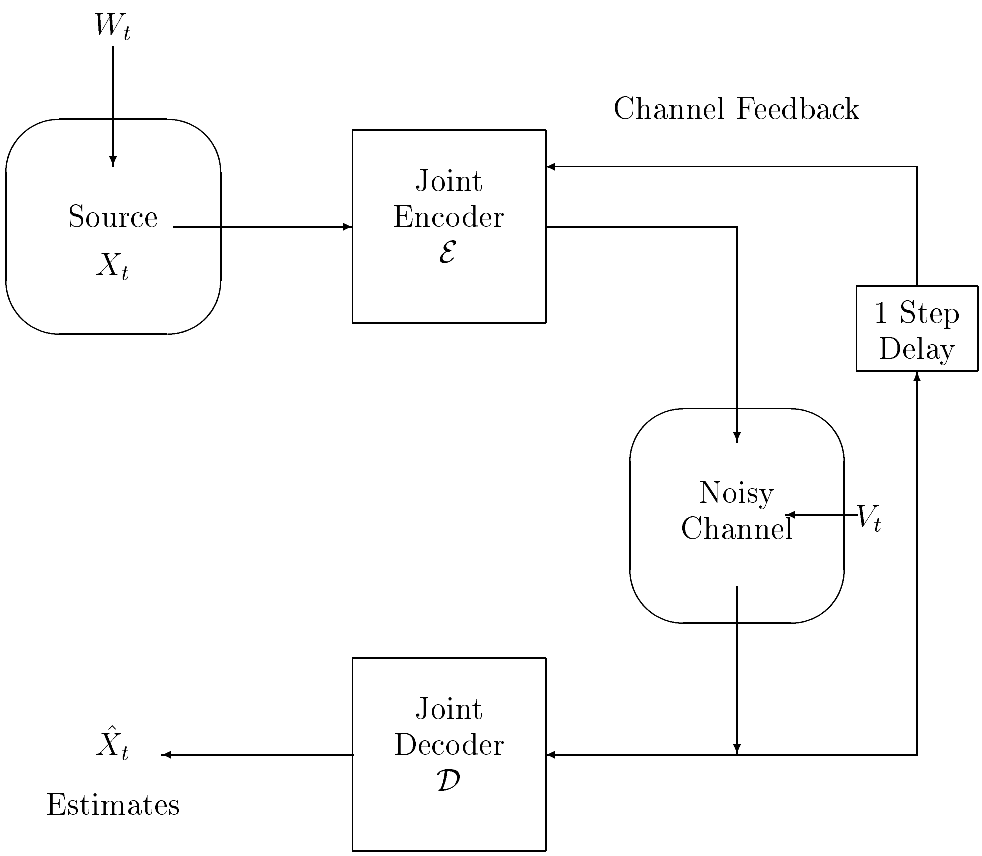

Figure 7-2: The estimation problem over a noisy channel with access to noiseless feedback

7.1.3 Equivalence with Estimation

By equivalence, we mean that $|\hat{X}_t - X_t|$ in the estimation problem of Figure 7-2 can be made to behave essentially like $|\tilde{X}_t|$ in the control problem of Figure 7-1. As we have seen in the definitions of stability, the key to evaluating the performance of a system is the complementary distribution function $P(|\tilde{X}_t| > x)$ in the control case or $P(|\hat{X}_t - X_t| > x)$ in estimation problems.

Theorem 7.1.1 *For a given noisy channel and bounded driving noise $-\frac{\Omega}{2} \leq W_t \leq \frac{\Omega}{2}$, if there exists an observer $\mathcal{O}$ and controller $\mathcal{C}$ for the controlled Markov source of Figure 7-1 that achieve $P(|\tilde{X}_t| > x) \leq f_t(x)$, then there exists a joint encoding system $\mathcal{E}$ with access to noiseless feedback and joint decoding system $\mathcal{D}$ for the estimation problem of Figure 7-2 that achieve*

$$P(|X_t - \hat{X}_t| > x) \leq f_{t+1}(Ax - \frac{\Omega}{2})$$

Similarly, if there exists a joint encoding system $\mathcal{E}$ with access to noiseless feedback and joint decoding system $\mathcal{D}$ for the estimation problem of Figure 7-2 which achieves $P(|X_t - \hat{X}_t| > x) \leq f'_t(x)$, then there exists an observer $\mathcal{O}$ (with access to noiseless channel feedback) and controller $\mathcal{C}$ for the controlled Markov source of Figure 7-1 which achieve

$$P(|\tilde{X}_t| > x) \leq f'_{t-1}(\frac{x - \frac{\Omega}{2}}{A})$$

Proof: We give constructions which satisfy the above bounds in Figures 7-3 and 7-4.

In order to construct a joint encoding system $\mathcal{E}$ from an observer $\mathcal{O}$ and controller $\mathcal{C}$, we follow Figure 7-3 and use:

$$\mathcal{E}_t(X^t, B^{t-1}) = \mathcal{O}(\left(X_j + \sum_{i=0}^{j-1} A^{j-1-i} U_i\right)_{j=0}^{t}, \left(\mathcal{C}(B^j)\right)_{j=0}^{t-1}, B_0^{t-1}) \tag{7.2}$$

In (7.2), we use superscripts to denote the final member of a sequence measured in *time*, rather than as a position in the sequence. The delay of 1 unit is considered to be of 1 sample in the discrete time channel.

The encoder above works because the input to the observer $\mathcal{O}$, is the virtual controlled source:

$$\begin{aligned} \tilde{X}_j &= X_j + \sum_{i=0}^{j-1} A^{j-1-i} U_i \\ &= \sum_{i=0}^{j-1} A^{j-1-i}(W_i + U_i) \\ &= A\tilde{X}_{j-1} + W_{j-1} + U_{j-1} \end{aligned}$$

which behaves exactly like the original controlled source. As a result, we know that $P(|\tilde{X}_t| > x) \leq f_t(x)$.

Out joint decoding system is similarly:

$$\hat{X}_t = \mathcal{D}_t(B^t) = -A^{-1} \sum_{i=0}^{t} A^{t-i} U_t \tag{7.3}$$

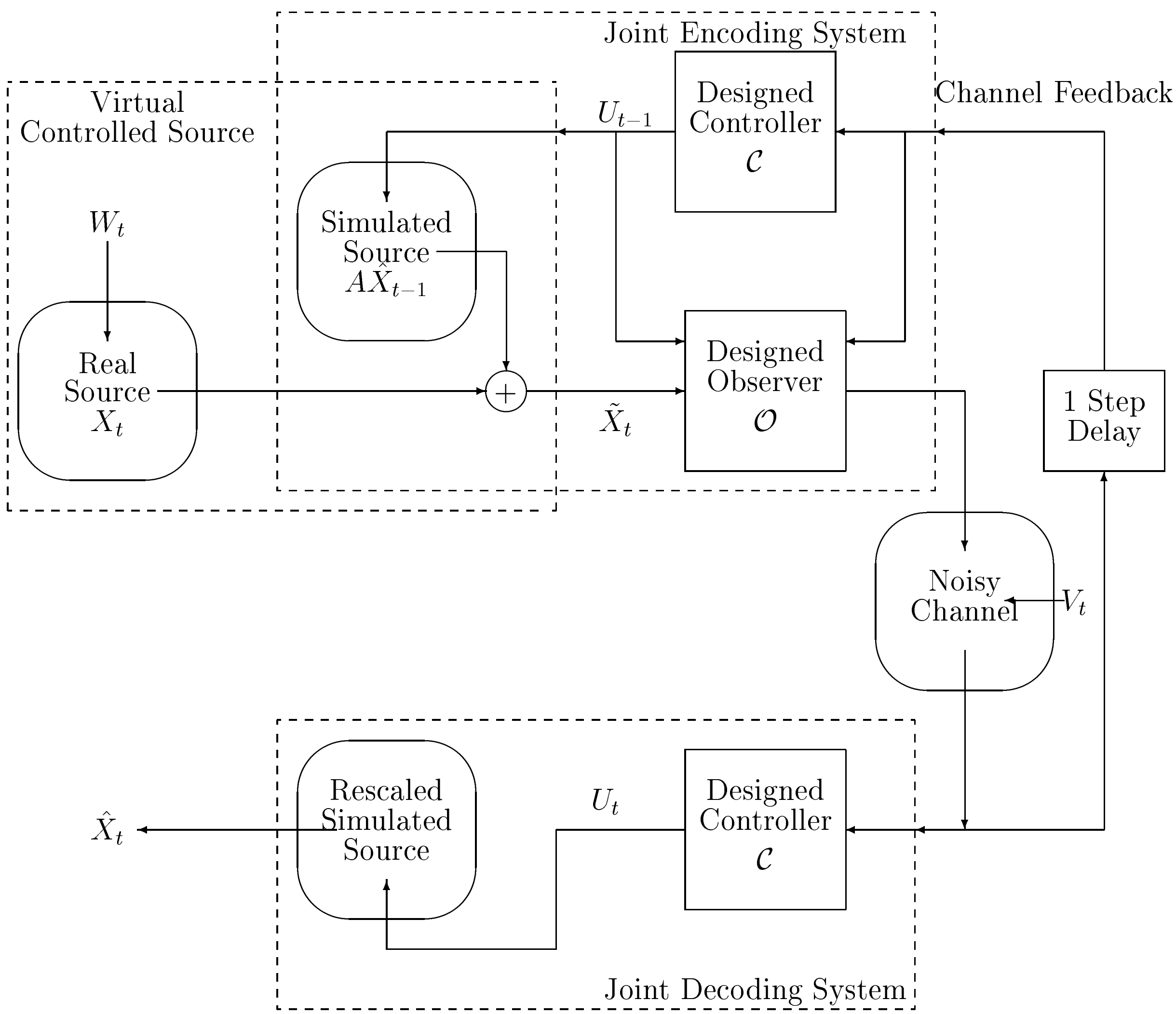

Figure 7-3: Block diagram showing how to construct an estimator from a control system

Putting things together we see that

$$\begin{aligned}\tilde{X}_t &= X_t - A\hat{X}_{t-1} \\ &= A(X_{t-1} - \hat{X}_{t-1}) + W_{t-1}\end{aligned}$$

And so:

$$\begin{aligned}P(|X_{t-1} - \hat{X}_{t-1}| > x) &= P(|A(X_{t-1} - \hat{X}_{t-1})| > Ax) \\ &= P(|A(X_{t-1} - \hat{X}_{t-1})| + W_{t-1} > Ax + W_{t-1}) \\ &\leq P(|A(X_{t-1} - \hat{X}_{t-1}) + W_{t-1}| > Ax - |W_{t-1}|) \\ &\leq P(|A(X_{t-1} - \hat{X}_{t-1}) + W_{t-1}| > Ax - \frac{\Omega}{2}) \\ &= P(|\tilde{X}_t| > Ax - \frac{\Omega}{2}) \\ &\leq f_t(Ax - \frac{\Omega}{2})\end{aligned}$$

which proves the first part of the theorem. The second part proceeds analogously using Figure 7-4 and consists of designing the controller as an estimator followed by the certainty-equivalent controller.

In order to construct the observer $\mathcal{O}$ from a joint encoder $\mathcal{E}$ and decoder $\mathcal{D}$, we follow Figure 7-4 and use:

$$\mathcal{O}(\tilde{X}^t, B^{t-1}) = \mathcal{E}(\left(\tilde{X}_j + A\mathcal{D}(B^{j-1})\right)_{j=0}^{t}, B^t) \tag{7.4}$$

and for the controller:

$$\mathcal{C}(B^t) = -A\left(\mathcal{D}(B^t) - A\mathcal{D}(B^{t-1})\right) \tag{7.5}$$

To see that (7.4) and (7.5) indeed define an appropriate system, we first need to verify that the input to the encoder $\mathcal{E}$ looks like an uncontrolled Markov Process. To see this:

$$\begin{aligned}\tilde{X}_j + A\mathcal{D}(B^{j-1}) &= A\tilde{X}_{j-1} + W_j - A\left(\mathcal{D}(B^{j-1}) - A\mathcal{D}(B^{j-2})\right) + A\mathcal{D}(B^{j-1}) \\ &= W_j + A\left(\tilde{X}_{j-1} + A\mathcal{D}(B^{j-2})\right) \\ &= \sum_{i=1}^{j} A^{j-i} W_i \\ &= X_j\end{aligned}$$

where we use induction to get the explicit sum from the recursive rule. To evaluate the performance of this setup, we rearrange terms slightly to get

$$\begin{aligned}\tilde{X}_j &= X_j - A\mathcal{D}(B^{j-1}) \\ &= A(X_{j-1} - \hat{X}_{j-1}) + W_j\end{aligned}$$

and so:

$$\begin{aligned}P(|\tilde{X}_t| > x) &= P(|A(X_{t-1} - \hat{X}_{t-1}) + W_t| > x) \\ &= P(|(X_{t-1} - \hat{X}_{t-1}) + \frac{W_t}{A}| > \frac{x}{A})\end{aligned}$$

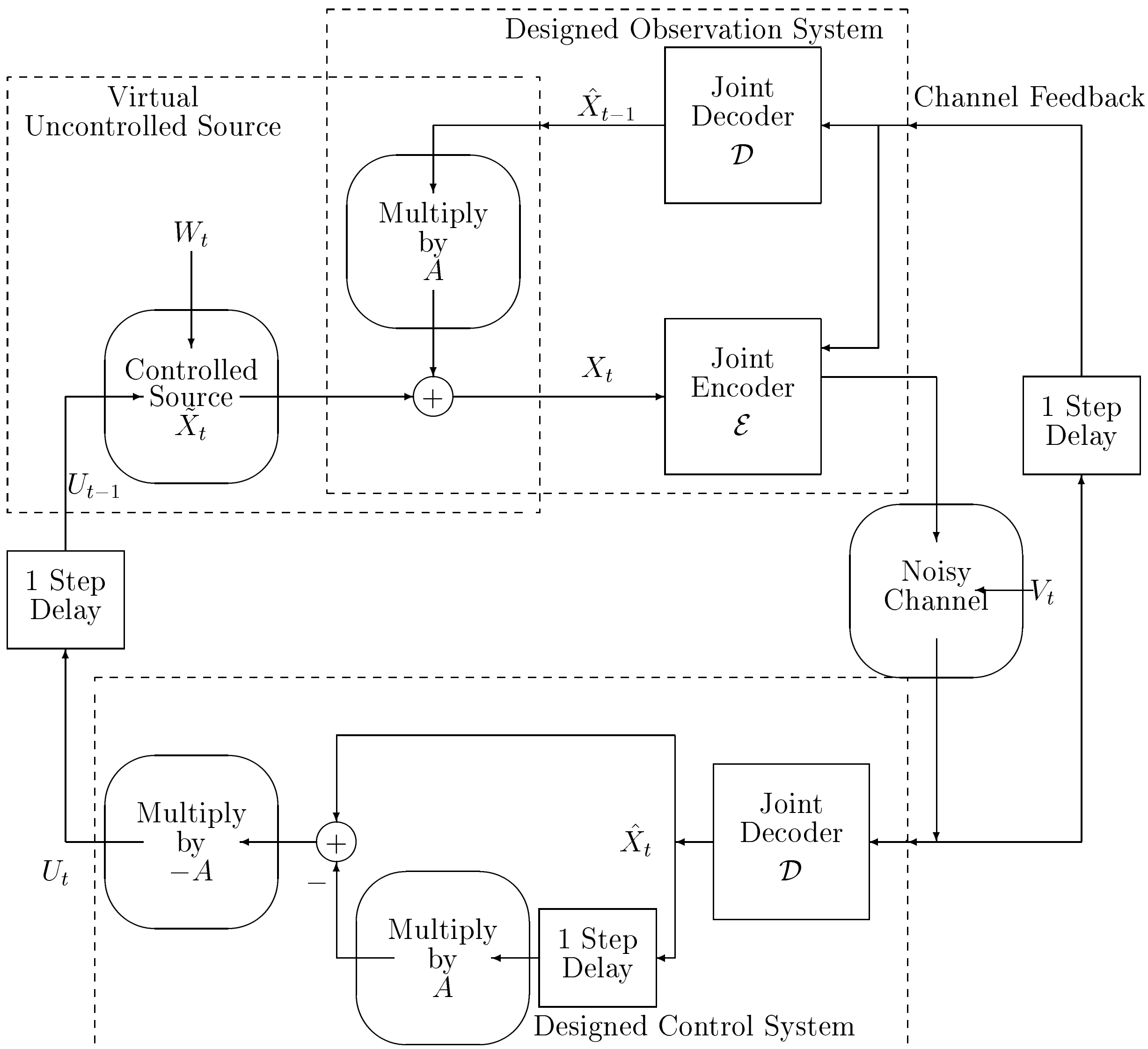

Figure 7-4: Block diagram showing how to construct a control system from an estimator

$$\leq P(|(X_{t-1} - \hat{X}_{t-1})| > \frac{x - |W_t|}{A})$$
$$\leq P(|(X_{t-1} - \hat{X}_{t-1})| > \frac{x - \frac{\Omega}{2}}{A})$$
$$\leq f'_{t-1}(\frac{x - \frac{\Omega}{2}}{A})$$

which proves the theorem. □

The assumption of bounded support on W_t was used in Theorem 7.1.1 to make the proof completely straightforward. However, it is easy to see that in the case where the driving noise $\{W_t\}$ is an i.i.d. random process, the essential point continues to hold even without the bounded support assumption since the additional effect of the noise can only spread out the distribution of $|\tilde{X}_t|$ relative to that of $|A(X_{t-1} - \hat{X}_{t-1})|$. Also, at the moment this result is given for scalar systems only. The general equivalence does continue to hold in vector problems, but requires some assumptions on controllability and observability which are trivially satisfied in the scalar case.

7.1.4 Information Transmission Theorem For Control

Theorem 7.1.1 as stated relates the complementary probability distribution functions of $|\tilde{X}_t|$ and $|X_t - \hat{X}_t|$ to each other and says that they are essentially the same, except for a rescaling by A and a possible shift by $\frac{\Omega}{2}$. In particular, this implies that if any finite moment of either can be kept bounded, the same moment can be kept bounded for the other one as well. This means that we can directly extend the direct part of our Information Transmission Theorem embodied in Corollary 6.2.1, to the control problem as well:

Corollary 7.1.1 *A controlled scalar discrete-time unstable linear Markov process with parameter a driven by bounded noise can be η-stabilized across a noisy channel with the observer having access to noiseless feedback if there is an $\epsilon > 0$ for which $C_{anytime}(\eta \log_2 a + \epsilon) > \log_2 a$ for the channel with noiseless feedback. In particular, if $C_{anytime}(2 \log_2 a + \epsilon) > \log_2 a$, then we can stabilize it in the mean-squared sense.*

Furthermore, since the proofs of the converse side to the Information Transmission Theorem also only need to use bounded driving signals to encode the information bits, our Theorem 7.1.1 also lets us extend Theorem 6.3.1 to the control problem giving us:

Corollary 7.1.2 *Given a noisy channel, if there exists an observer (possibly with access to noiseless feedback of the channel output) and controller $(\mathcal{O}, \mathcal{C})$ which stabilizes a scalar discrete-time controlled linear Markov processes $\{\tilde{X}_t\}$ with parameter A driven by any bounded noise signals $-\frac{\Omega}{2} \leq W_t \leq \frac{\Omega}{2}$ so that:*

$$P(|\tilde{X}_t| \geq \Delta) \leq f(\Delta)$$

then for all $\epsilon > 0$ the channel with noiseless feedback has $C_{anytime}(\tilde{f}) > \log_2 A - \epsilon$ where $\tilde{f} = f(\delta' 2^{d \log_2 A} - \frac{\Omega}{2})$ for some constant $\delta' > 0$.

In particular, if the original system can be η-stabilized, then $\tilde{f}(d)$ can be made to be like $2^{-(\eta \log_2 A) d}$ giving us an α-anytime-capacity $C_{anytime}(\eta \log_2 A) > \log_2 A - \epsilon$ for the channel with access to noiseless feedback.

Similar extensions are possible to Corollaries 6.4.1 and 6.4.2 making them applicable in the control context as well. This tells us that the necessity parts of the separation theorem also extend to control problems — anytime capacity is necessary to evaluate a channel for closed-loop control as well as for estimation. It is this control interpretation that is much more important since in the real world, unstable processes usually exist in an "interactive" or "responsive" setting. Control theory is just the mathematical formalism we have for dealing rigorously with such settings.

7.2 Specific Channels With Feedback

To illustrate the value of the conceptual equivalence between control and estimation, we will now examine two important channels with encoders having access to noiseless feedback: the binary erasure channel and power-constrained additive noise channels, particularly the AWGN channel. For both of these, control strategies to stabilize the system are obvious by inspection and so we can use Corollary 7.1.2 to get anytime capacities for these channels.

7.2.1 Erasure Channel

To avoid complication, we will assume that the channel sampling time is the same as the source sampling time in Figure 7-1 where the noisy channel is a binary erasure channel with erasure probability $e > 0$. We consider the case where the driving noise $\{W_t\}$ is arbitrary, but known to be bounded so that: $-\frac{\Omega}{2} \leq W_t \leq \frac{\Omega}{2}$.

Now, consider a hypothetical external observer located at the controller. This observer knows the encoder and the control law being used, but can only see the outputs of the channel. Suppose that before seeing channel output b_t the observer knew that $\tilde{X}_t \in \Upsilon_t$. Now, imagine that an erasure occurs and $b_t = \emptyset$. The observer has received no information and can thus only update the uncertainty to conclude $X_{t+1} \in \{Ax + w + U_{t+1} | x \in \Upsilon_t, w \in [-\frac{\Omega}{2}, \frac{\Omega}{2}]\}$. The observer cannot distinguish between the points within this expanded set since the uncertainty comes from the fact that W^t are arbitrary.

Optimal Control and Observation

Since the goal is to keep the maximal value for $\tilde{X}_t$ small, we can focus on Λ_t from (7.1). As such, the choice of control is immediately clear. The control signal U_t should be chosen to minimize Λ_{t+1}. Thus the optimal $-U_t$ is the midpoint of the smallest interval that can contain the possible values for X_{t+1} without control. The result of such a control is to make the post-control uncertainty for X_{t+1} lie within an interval centered on the origin.

The next thing to notice is that because of the noiseless feedback, the observer knows everything the controller does. As such, its goal is to make these intervals as small as possible by the choice of bits sent. Every bit a_t sent by the observer is there to tell the controller which of two intervals contains the value for $\tilde{X}_t$. The intervals themselves must be determinable from the previous received values and it is immediately clear that the optimal choice for the splitting must be a partition with no overlap between the two intervals. Any overlap would be suboptimal since it would make at least one of the intervals longer than it has to be.

Now, let Y_t represent the size of the controller's uncertainty interval about the state of the system. This is a random variable depending only on the realization of the channel

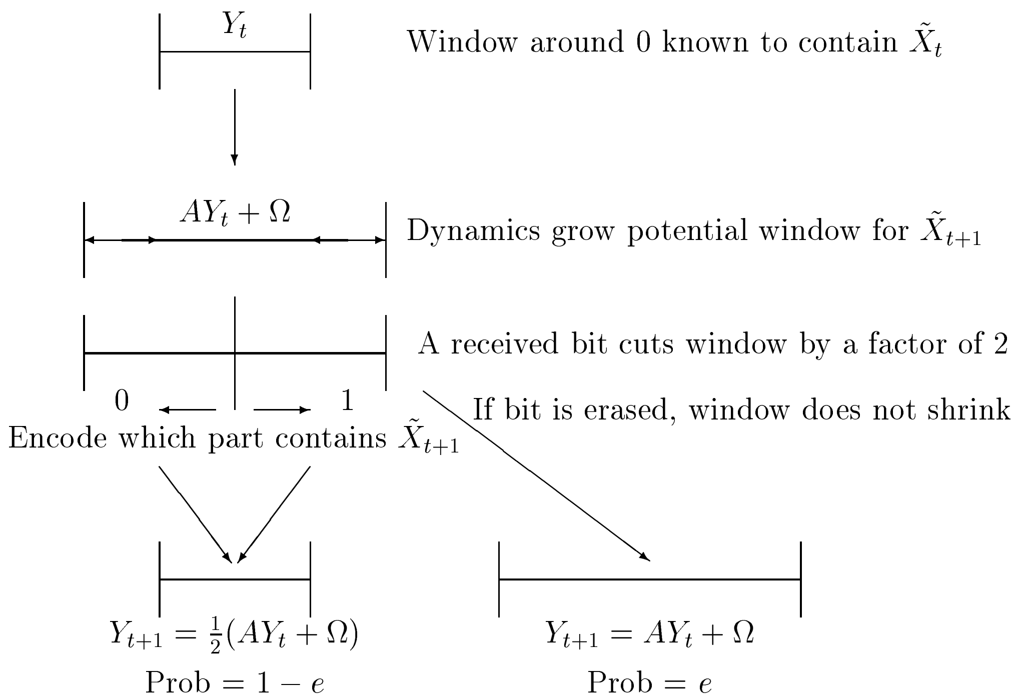

Figure 7-5: The evolution of the uncertain interval on the erasure channel with noiseless feedback.

noise. When there is an erasure, it grows according to the dynamics of the system:

$$Y_{t+1} = AY_t + \Omega$$

If a bit does get through, it depends on how the interval is divided into two parts. If the division is done in a deterministic way, the driving noise could always conspire to pick the larger interval. Hence, the optimal thing to do with deterministic observers is to divide the interval into two equal sized pieces.[1] Thus, if there is no erasure, the best we can do is:

$$Y_{t+1} = \frac{A}{2}Y_t + \frac{\Omega}{2}$$

With the known initial condition of $Y_0 = 0$, $\{Y_t\}$ is now a well defined random process regardless of the realization of the arbitrary driving noise, even though the actual bits transmitted depend on that noise. The evolution of Y_t is depicted in Figure 7-5. At every time step, a new Ω is added in while all the previous Ω's get multiplied by a random factor. Write Y_t out as a sum: $Y_t = \sum_{n=0}^{t-1} \Omega A^n F_n$ where the F_n are correlated random variables which express how many of the last n channel transmissions have gotten through. In particular $F_n = \prod_{k=t-n}^{t} (\frac{1}{2})^{I_k}$ where I_k is the indicator for the event that the k-th transmission was not erased. Individually, the probability distribution of the F_n distributions is easy to see: $F_n = \frac{1}{2^i}$ with probability $\frac{n!(1-e)^i e^{n-i}}{i!(n-i)!}$.

[1]Even with probabilistic observers sharing common randomness with the controller, we can do no better on average since $(t^2 + (1-t)^2) = (1 - 2t + 2t^2)$ achieves its unique minimum at $t = \frac{1}{2}$.

Stability

It should be clear that η-stability of the system is equivalent to $E[Y_t^\eta] \leq K$ for all t. For a moment, concentrate on $\eta = 2$ for convenience. We can study the asymptotic properties of Y_t^2 by formally considering:

$$E[Y_\infty^2] = E\left[\Omega^2 \sum_{j=0}^{\infty}\sum_{n=0}^{\infty} A^{j+n} F_j F_n\right]$$

Define N_k^j as the random variable for which $F_{j+k} = F_j N_k^j$. Then group terms:

$$\begin{aligned} E[Y_\infty^2] &= E\left[\Omega^2 \sum_{j=0}^{\infty} A^{2j} F_j^2 (1 + 2\sum_{k=1}^{\infty} A^k N_k^j)\right] \\ &= \Omega^2 \sum_{j=0}^{\infty} A^{2j}(E[F_j^2] + 2\sum_{k=1}^{\infty} A^k E[F_j^2 N_k^j]) \end{aligned}$$

F_j and N_k^j are independent by construction, and hence expectations distribute over them. Moreover, by the memorylessness of the channel, it is clear that $E[N_k^j]$ does not depend on j at all, and in fact $E[N_k^j] = E[F_j]$. So we get:

$$\begin{aligned} &\frac{E[Y_\infty^2]}{\Omega^2} \\ = &\left(\sum_{j=0}^{\infty} A^{2j} E[F_j^2]\right)\left(1 + 2\sum_{k=1}^{\infty} A^k E[F_k]\right) \\ = &\left(\sum_{j=0}^{\infty} A^{2j} \sum_{i=0}^{j} \frac{j!(1-e)^i e^{j-i}}{i!(j-i)!4^i}\right) \\ &\left(1 + 2\sum_{k=1}^{\infty} A^k \sum_{l=0}^{k} \frac{k!(1-e)^l e^{k-l}}{l!(k-l)!2^l}\right) \\ = &\left(1 + \sum_{j=1}^{\infty} (A^2)^j e^j \sum_{i=0}^{j} \frac{j!}{i!(j-i)!}(\frac{1-e}{4e})^i\right) \\ &\left(1 + 2\sum_{k=1}^{\infty} A^k e^k \sum_{l=0}^{k} \frac{k!}{l!(k-l)!}(\frac{1-e}{2e})^i\right) \\ = &\left(1 + \frac{A^2(1+3e)}{4 - A^2(1+3e)}\right)\left(1 + \frac{2A(1+e)}{2 - A(1+e)}\right) \end{aligned}$$

More importantly, this is only valid if $A < \frac{2}{\sqrt{1+3e}} < \frac{2}{1+e} < 2^{1-e}$. Otherwise, the sums diverge. Since we did not have to make any approximations, this actually tells us the maximal A for which we can hope to mean-square stabilize the system. If we repeat the above argument with general $\eta > 0$, it turns out that the dominant term is:

$$\left(\sum_{j=0}^{\infty} A^{\eta j} \sum_{i=0}^{j} \frac{j!(1-e)^i e^{j-i}}{i!(j-i)!(2^\eta)^i}\right)$$

$$= \quad 1+\sum_{j=1}^{\infty} A^{\eta j} e^j \sum_{i=0}^{j} \frac{j!\left(\frac{1-e}{2^\eta e}\right)^i}{i!(j-i)!}$$

$$= \quad 1+\frac{(2A)^\eta(e+2^{-\eta}-e2^{-\eta})}{2^\eta(1-A^\eta e)-A^\eta(1-e)}$$

where the final simplification is calculated using Maple and Mathematica. It only converges if

$$A < \frac{2}{(1+e(2^\eta-1))^{\frac{1}{\eta}}} \tag{7.6}$$

Once again, to get the condition in (7.6), we did not have to make any approximations. Given that the observer and control strategy used are optimal, this condition is both necessary and sufficient for stability giving us the following theorem:

Theorem 7.2.1 *To η-stabilize the unstable Markov controlled source with parameter A and bounded driving noise over a binary erasure channel with erasure probability e, it is necessary that we satisfy (7.6). This condition is also sufficient if the observer has access to noiseless feedback.*

Anytime Capacity

To calculate the α-anytime capacity for the binary erasure channel with noiseless feedback, we just need to combine Theorem 7.2.1 with Corollary 7.1.2 which tells us that the condition for stability is that $C_{\text{anytime}}(\eta \log_2 A) \geq \log_2 A$. Plugging in the result from (7.6) gives us:

$$C_{\text{anytime}}(\eta \log_2 \left(\frac{2}{(1+e(2^\eta-1))^{\frac{1}{\eta}}}\right)) \geq \log_2 \left(\frac{2}{(1+e(2^\eta-1))^{\frac{1}{\eta}}}\right)$$

Simplifying and recognizing that (7.6) was both necessary and sufficient gives us:

Theorem 7.2.2 *For the binary erasure channel with encoders having access to noiseless feedback,*

$$C_{anytime}\left(\eta - \log_2(1+e(2^\eta-1))\right) = 1 - \frac{1}{\eta}\log_2\left(1+e(2^\eta-1)\right)$$

where $\eta > 0$

It is interesting to notice that this has the asymptotes we would expect: $C_{\text{anytime}}(0) = 1-e$ and $C_{\text{anytime}}(\alpha \geq -\log_2 e) = 0$. Also, the result is stated for a unit inter-sample time between channel uses. If the sampling time is different, a simple scaling will translate the result.

7.2.2 Additive Noise Channels

Once again, we assume that the channel sampling time is the same as the source sampling time in Figure 7-1 where the noisy channel is an additive white Gaussian noise channel with zero mean unit variance noise and a power constraint of P. We also assume that the driving noise $\{W_t\}$ is i.i.d. and has zero mean and unit variance. It turns out that for most of this section, we will only use that fact that the additive noise is zero mean, unit variance, and white. The Gaussianity will only come in later.

Control and Observation

Since this is a linear system with additive noise, it can be viewed as a classical partially observed control problem[70], except that we get to design the observer[3]. Our approach will be to use a scalar gain β for the observer and then another gain γ for the controller. Thus:

$$Y_t = \beta \tilde{X}_t \tag{7.7}$$

where Y_t is the channel input and $\tilde{X}_t$ is the system state.

$$U_t = -A\gamma B_t \tag{7.8}$$

where U_t is the control signal and $B_t = Y_t + V_t$ the noisy observation.

Satisfying the power constraint is then a matter of insuring that $E[Y_t^2] \leq P$ which automatically implies mean-squared stability of $\tilde{X}_t$ by (7.7). Plugging in both (7.7) and (7.8) into the definition of the system gives us:

$$\tilde{X}_{t+1} = A(1-\gamma\beta)\tilde{X}_t + (W_{t+1} - A\gamma V_t) \tag{7.9}$$

By inspection, the driving noise for this system $(W_{t+1} - A\gamma V_t)$ is i.i.d. with zero mean and variance $1 + A^2\gamma^2$. The condition for closed loop stability is that $A(1-\gamma\beta) < 1$ and so $\gamma\beta > 1 - \frac{1}{A}$. In such a situation, the $\tilde{X}$ asymptotically behaves like:

$$\tilde{X} = \sum_{j=0}^{\infty} W_{j+1}(A(1-\gamma\beta))^j - \sum_{j=0}^{\infty} A\gamma V_j (A(1-\gamma\beta))^j \tag{7.10}$$

Assuming that, we can calculate the asymptotic variance of $\tilde{X}$ as:

$$E[\tilde{X}^2] = \frac{1 + A^2\gamma^2}{1 - A^2(1-\gamma\beta)^2}$$

Now assume that the power constraint is satisfied with equality, $E[Y_t^2] = P$, and set $\beta\gamma = \frac{P}{P+1}$ inspired by the standard minimum squared error estimator. This only makes sense if stability is maintained:

$$\begin{aligned} 1 - \frac{1}{A} &< \gamma\beta \\ &= \frac{P}{P+1} \\ &= 1 - \frac{1}{1+P} \end{aligned}$$

and so $A < P + 1$. Furthermore, the power constraint itself has to be met and so:

$$\begin{aligned} P &= E[Y^2] \\ &= \beta^2 E[\tilde{X}^2] \\ &= \frac{\beta^2 + A^2(\beta\gamma)^2}{1 - A^2(1-\gamma\beta)^2} \\ &= \frac{\beta^2 + A^2(\frac{P}{P+1})^2}{1 - A^2(\frac{1}{P+1})^2} \end{aligned}$$

$$= \frac{\beta^2(P+1)^2 + (AP)^2}{(P+1)^2 - A^2}$$

cross multiplying gives us:

$$\begin{aligned}
\Big((P+1)^2 - A^2\Big) P &= \beta^2(P+1)^2 + (AP)^2 \\
\Big((P+1)^2 - A^2(P+1)\Big) P &= \beta^2(P+1)^2 \\
\Big((P+1) - A^2\Big) P &= \beta^2(P+1) \\
\Big((P+1) - A^2\Big) &= \frac{\beta^2(P+1)}{P}
\end{aligned}$$

which can be satisfied with appropriate choice of $\beta = \sqrt{P - \frac{A^2 P}{P+1}}$ as long as $(P+1) - A^2 > 0$ or:

$$A < \sqrt{P+1} \tag{7.11}$$

which also implies closed-loop stability since $A < \sqrt{P+1} < P+1$.

This is clearly a sufficient condition for mean-squared stability. To see that it also suffices for general η-stability, we notice that (7.10) tells us that the system behaves as the convergent geometric sum of two sets of i.i.d. random variables $\{W_t\}$ and $\{V_t\}$. As long as both of them have finite η-moments, so will their convergent geometric sums. If the system driving noise is bounded to lie inside $[-\frac{\Omega}{2}, \frac{\Omega}{2}]$, we know from (7.10) that its contribution to $\tilde{X}$ is guaranteed to be within $[-M, M]$ where:

$$\begin{aligned}
M &= \frac{\Omega}{2}\frac{1}{1 - A(1-\gamma\beta)} \\
&\leq \frac{\Omega}{2(1 - \frac{A}{P+1})} \\
&= \frac{\Omega(P+1)}{2(P+1-A)} \\
&\leq \frac{\Omega(\sqrt{P+1})}{2(\sqrt{P+1}-1)}
\end{aligned}$$

Thus we have:

Theorem 7.2.3 *To η-stabilize the unstable Markov controlled source with parameter A and bounded driving noise over a additive white noise channel with zero mean and unit variance (and bounded η-moment) with power constraint P, it is sufficient that (7.11) be satisfied.*

Anytime and Zero-error Capacity

To calculate a lower bound on the anytime capacity for the power constrained additive noise channel with noiseless feedback, we will again combine Theorem 7.2.3 with Corollary 7.1.2. In this case, we notice that if the system driving noise $\{W_t\}$ has bounded support, then (7.10) tells us that the tail of $\tilde{X}$ is determined entirely by the convergent geometric sum $\sum_{j=0}^{\infty} A\gamma V_j (A(1-\gamma\beta))^j$ of the channel's additive noise $\{V_t\}$. There are two cases of particular interest.

The first is when V_t too has bounded support. In that case, there exists a constant $M > 0$ depending on P and the bounds for W_t and V_t such that: $P(|\tilde{X}| > M) = 0$. So, we can construct a function

$$f(x) = \begin{cases} 1 & \text{if } x \leq M \\ 0 & \text{otherwise} \end{cases}$$

so that $P(|\tilde{X}_t| > x) \leq f(x)$. An immediate application of Corollary 7.1.2 tells us that for this channel with noiseless feedback, we get : $C_{\text{anytime}}(\tilde{f}) \geq \log_2 A$ where $\tilde{f} = f(\delta' 2^{d \log_2 A} - \frac{\Omega}{2})$ for some constant $\delta' > 0$. Plugging in (7.11) gives us: $C_{\text{anytime}}(\tilde{f}) \geq \frac{1}{2}\log_2(1+P)$ and there exists an M' for which $\tilde{f}(d) = 0$ if $d > M'$ giving us the following theorem:

Theorem 7.2.4 *For an additive white noise channel with bounded unit variance noise and power constraint P, the zero error capacity with access to noiseless feedback is bounded by:*

$$C_0 \geq \frac{1}{2}\log_2(1+P)$$

Notice that the lower bound to zero-error capacity in Theorem 7.2.4 does not depend on the bound Ω for the noise support! It is a function only of the power constraint relative to the average noise power. The bound on the noise only effects the delay we must be willing to tolerate before we can be absolutely confident of our estimates.

The next case of interest is that of unit variance Gaussian noise. In that case, the fact that the sum of i.i.d. Gaussians is Gaussian tells us that the $\sum_{j=0}^{\infty} A\gamma V_j (A(1-\gamma\beta))^j$ is a zero mean Gaussian with variance:

$$\begin{aligned} \sigma_A^2 &= \sum_{j=0}^{\infty} A^2\gamma^2 (A^2(1-\gamma\beta)^2)^j \\ &= \frac{A^2\gamma^2}{1 - A^2(1-\gamma\beta)^2} \\ &= \frac{A^2 \frac{P}{(1+P)^2 - A^2(1+P)}}{1 - A^2(\frac{1}{1+P})^2} \\ &= \frac{PA^2(1+P)}{((1+P)^2 - A^2)(1+P-A^2)} \end{aligned}$$

which tends to infinity as we approach the limit of $A = \sqrt{1+P}$. Thus, for a suitably large constant M'', the tail of the distribution dies at least as fast as

$$f(x) = M'' e^{-\frac{x^2}{2\sigma_A^2}}$$

An immediate application of Corollary 7.1.2 tells us that for this channel with noiseless feedback, we get: $C_{\text{anytime}}(\tilde{f}) > \log_2 A - \epsilon$ where $\tilde{f} = f(\delta' A^d - \frac{\Omega}{2})$ for some constant $\delta' > 0$. Setting $R < \log_2 A \leq \frac{1}{2}\log_2(1+P)$, we know that $2^{Rd} \leq A^d$. This means that the probability of error goes to zero double exponentially at least as fast as

$$e^{-\frac{\delta'^2}{2\sigma_A^2} 4^{Rd}} \tag{7.12}$$

where d is the delay we are willing to accept gets larger. Since a double exponential is faster than any single exponential, we have proved the following:

Theorem 7.2.5 *For the AWGN channel with power constraint P and encoders having access to noiseless feedback,*

$$C_{anytime}(\alpha) = \frac{1}{2}\log_2(1+P)$$

for any $\alpha > 0$

The equality holds since we know that anytime capacity cannot exceed the regular Shannon capacity which is also $\frac{1}{2}\log_2(1+P)$ for the AWGN channel with or without feedback.

Comparison to Schalkwijk and Kailath

This double exponential convergence is reminiscent of the result of Schalkwijk and Kailath[59]. The difference is that we do not have to assume any block-length *a priori* and can deliver the performance simultaneously over all sufficiently large delays, eventually getting every bit correct. Thus, the comparison between the rate at which the probability of error goes to zero for the two different schemes is like comparing apples to oranges. Even so, the results are surprising.

From [59], we have that the probability of error for the Schalkwijk and Kailath scheme is

$$P_e \leq \sqrt{\frac{2}{3\pi}} e^{-N(C-R)-\frac{3}{2}e^{2N(C-R)}} \tag{7.13}$$

where C is the capacity in nats per channel use, R is the rate of transmission in nats per channel use, and N is the block-length used. Neglecting constants and other smaller effects, the dominant term in (7.13) is a double exponential of the form $K_1^{-K_2^N}$ where $K_1 = e$ and $K_2 = e^{2(C-R)}$. Notice that as the rate gets closer to capacity, it is the inner exponential base that gets closer to 1 while the outer exponential base stays fixed at e.

To compare, we express the rate at which the probability of error goes to zero as $K_3^{-K_4^d}$ where d is the delay at which the decoder chooses to decode. (7.12) gives us $K_3 = e^{\frac{\delta'^2}{2\sigma_A^2}}$ and $K_4 = 4^R$ where R is the rate in bits per channel use. Since δ' varies linearly in ϵ_1 where $R = \log_{2+\epsilon_1} A$, it tends to zero as the rate gets closer to capacity. Similarly, σ_A^2 tends to infinity as we get closer to capacity. Thus, as the rate of transmission approaches capacity, for our scheme the outer exponential base gets closer to 1 while the inner exponential base tends upwards to a constant 4^C (where C is the capacity in bits per channel use). The situation is qualitatively different.

The natural question is, if we fix the rate $R < C$, which scheme is asymptotically better in the limit of large delays? The answer is immediate if we look at logarithms. The Schalkwijk and Kailath scheme has the $-\log$ of the probability of error growing as $O(K_2^N)$ while ours has the $-\log$ of the probability of error growing as $O(K_4^d)$. Near capacity, K_2 is close to 1 while K_4 can be substantially bigger. For large d and N, our scheme's faster inner exponent completely dominates over whatever penalty it suffers in the constants. This advantage is larger the closer we are to capacity.

7.3 Discussion

In this chapter, we have established the connection between the estimation problems we have been discussing in previous chapters and their corresponding control versions where feedback is over a noisy link. We show that the converse theorems establishing the necessity of sufficient anytime capacity for estimation also apply to the control problem of stabilizing a simple scalar system. This correspondence is then used to evaluate the anytime capacity for the binary erasure and AWGN channels when the encoder has access to noiseless feedback. In particular, we are able to show the double-exponential convergence of errors to zero for the AWGN case.

The simple control problems considered here are only the beginning of what is hoped will be a long productive interaction between information theory and control. The results need to be extended to the vector valued plant case and imperfect observations. Also, existing results on codes for channels with feedback [34] have control implications that should be explored. There are also intriguing possibilities of viewing a stochastically disturbed plant itself as a communication channel between the controller and the observer [55] that need to be fleshed out.

More speculatively, the implications of our ideas of anytime capacity need to be explored in the context of hierarchical and distributed control systems. The "delay universality" of an anytime encoder might be a useful property in a control system which operates at different time scales. The same transmission could conceivably be used to provide accurate estimates to a slower monitoring controller while still providing useful low latency information to a real-time regulator.

Chapter 8

"Quality of Service"

Our Information Transmission theorem (Corollary 6.2.1 and Theorem 6.3.1) showed that transmitting unstable Markov processes needs more from a communication link than just a given bit-rate. It requires that the α parameter at that bit rate also be high enough. This parameter α has the appearance of a fundamental "quality of service requirement" distinct from the rate of transmission. The use of the parameter α as a way of evaluating QoS has the advantage that the definition is not dependent on the specific digital nature of the channel. In principle, it should be applicable to wireless and other channels as well. It has a firm theoretical justification since we have converse theorems showing that it is necessary for the transmission and control of unstable Markov processes.

When we consider the issue of multiple streams sharing a single resource, simple time-sharing is the first and most obvious approach. It is almost a tautology that bit-rate is additive with time-sharing and so if that is the only QoS parameter, the situation is trivial since the resource can be characterized by a single number: how much rate it delivers. It is only when there is another QoS parameter that things can get interesting. The second parameter lets us formalize different trade-offs that come from different ways of sharing the resource, generally called "differentiated service."

In this chapter, we justify the QoS interpretation of the α parameter in anytime capacity by considering a simple vector source and the problem of estimating it in a mean-squared sense across a binary erasure channel with feedback. We introduce the basic problem setup and show that it is impossible to reliably transmit the source over the channel in a mean-squared sense if we insist on treating all bits alike in the reliable transmission layer. Finally, we show that providing differentiated service (with different α parameters for different streams) at the reliable transmission layer can allow us to achieve finite end-to-end mean squared error. This example shows that we do not need to specify an end-to-end delay constraint or any other user-level QoS requirement *a priori*. The need for QoS requirements and differentiated service can emerge from the nature of the source and the distortion measure.

8.1 A Simple Vector Problem

We consider a particular vector source and the problem of transmitting it with finite mean squared error across a binary erasure channel with noiseless feedback as shown in Figure 8-1.

The specific vector source we consider has $X_t \in \Re^5$, a bound on the driving noise

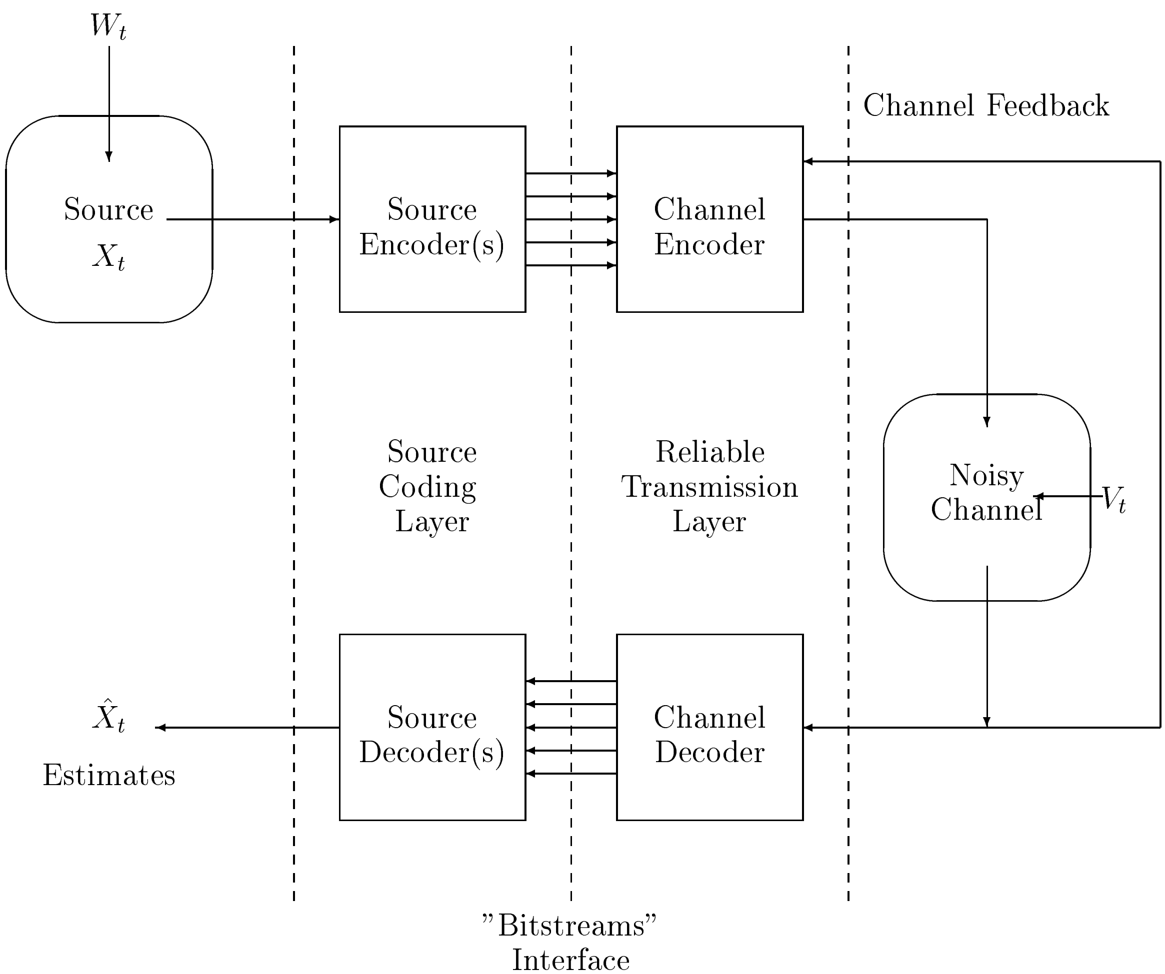

Figure 8-1: The vector estimation problem over a noisy channel with noiseless feedback and enforced layering

$\|W_t\|_\infty \leq \frac{\delta}{2}$, and known initial condition $X_0 = 0$. It is defined by the matrix of dynamics

$$A = \begin{pmatrix} 1.178 & 0 & 0.04 & 0 & 0.04 \\ 1.08 & 1.058 & 0.36 & 0 & 0.36 \\ 0.36 & 0 & 1.178 & 0 & 0.12 \\ -0.12 & 0 & -0.04 & 1.058 & -0.04 \\ -0.12 & 0 & -0.04 & 0 & 1.018 \end{pmatrix} \tag{8.1}$$

In our simple example, we use a binary erasure channel with $e = 0.27$.

8.1.1 Transform Coding

Looking at the source process in transformed coordinates is often of value in lossy coding.[35] In our case, it is illuminating to consider the transformed variable $\tilde{X}_t = TX_t$ where

$$T = \begin{pmatrix} 3 & 0 & 1 & 0 & 1 \\ 0 & 1 & -2 & 0 & 3 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

In that case, the transformed dynamics are given by a new $\tilde{A} = TAT^{-1}$ matrix:

$$\tilde{A} = \begin{pmatrix} 1.258 & 0 & 0 & 0 & 0 \\ 0 & 1.058 & 0 & 0 & 0 \\ 0 & 0 & 1.058 & 0 & 0 \\ 0 & 0 & 0 & 1.058 & 0 \\ 0 & 0 & 0 & 0 & 1.058 \end{pmatrix} \tag{8.2}$$

The initial condition remains $\tilde{X}_0 = 0$ and the new driving noise $\tilde{W}_t$ has a larger bound given by $\|\tilde{W}_t\|_\infty \leq 3\delta$. It is also clear that having an encoder/decoder pair which achieves a finite mean squared error between the original $\{X_t\}$ and $\{\hat{X}_t\}$ processes is equivalent to having a pair which achieves a finite mean squared error between the transformed $\{\tilde{X}_t\}$ process and its corresponding transformed reconstruction. Alternatively, we can think of the transformed system as being the underlying one. In that case, it can be considered a simple model for situations in which a single resource (the noisy link) needs to be shared "fairly" among multiple users so that everyone can get their jobs done.

Furthermore, being able to achieve a finite mean squared error with any finite end-to-end delay implies that we are able to achieve a finite mean squared error with no delay. This is because we could use an estimate for X_{t-d} to give an estimate for X_t by simply premultiplying it by A^d. After all,

$$\begin{aligned} & E\left[\|X_t - A^d\hat{X}_{t-d}\|^2\right] \\ = \; & E\left[\|(A^d X_{t-d} + \sum_{i=1}^{d} A^{i-1}W_{t-i}) - A^d\hat{X}_{t-d}\|^2\right] \\ = \; & E\left[\|A^d(X_{t-d} - \hat{X}_{t-d}) + \sum_{i=1}^{d} A^{i-1}W_{t-i}\|^2\right] \end{aligned}$$

$$
\begin{aligned}
&= E\left[\|A^d(X_{t-d}-\hat{X}_{t-d})\|^2 + \|\sum_{i=1}^{d} A^{i-1}W_{t-i}\|^2 + 2(A^d(X_{t-d}-\hat{X}_{t-d}))'(\sum_{i=1}^{d} A^{i-1}W_{t-i})\right] \\
&\leq E\left[\|A^d(X_{t-d}-\hat{X}_{t-d})\|^2\right] + E\left[\|\sum_{i=1}^{d} A^{i-1}W_{t-i}\|^2\right] \\
&\quad +2E\left[\|A^d(X_{t-d}-\hat{X}_{t-d})\|\|\sum_{i=1}^{d} A^{i-1}W_{t-i}\|\right]
\end{aligned}
$$

For any finite delay d, the three terms on the right hand side are all bounded for all t if we can achieve finite mean squared error. Hence the sum is bounded as well.

8.1.2 Fundamental Requirements

A necessary condition for achieving finite mean squared error for a vector valued process is getting a finite mean squared error for all the components. In the transformed domain, the dynamics of the components of our example are like those of five separate scalar unstable Markov sources. One of these is fast, and the four others are not as fast. Therefore, our Theorem 6.3.1 applies and we need the channel to satisfy both:

$$
\begin{aligned}
C_{\text{at}}(2\log_2(1.258)) &> \log_2(1.258) \\
C_{\text{at}}(2\log_2(1.058)) &> \log_2(1.058)
\end{aligned}
$$

Moreover, our work on sequential rate distortion[63] tells us that the total rate must be larger than the the sum of the logs of the unstable eigenvalues, giving us the additional condition that:

$$C > \log_2(1.258) + 4\log_2(1.058)$$

Thanks to Theorem 7.2.2, it is easy to check that all these requirements are satisfied individually by the binary erasure channel with erasure probability $e = 0.27$. (See Figure 8-2) However as stated, these are individually only necessary conditions, not sufficient ones. In a sense, we need them all satisfied simultaneously.

8.2 Treating All Bits Alike

By using the technique from Theorem 4.1.1, it is easy to construct recursive source codes for each of the five components of the transformed source. As long as we allow ourselves $R_1 > \log_2(1.258)$ bits per unit time for encoding the first component and $R_{2,3,4,5} > \log_2(1.1)$ bits per unit time time on each of the others, we can achieve finite mean squared error assuming no errors in transmitting these bits.

If we follow a strictly layered strategy and then require that all these bits be treated identically by the reliable transmission layer as shown in Figure 8-3, we come up against a problem. For an erasure channel with $e = 0.27$, Theorem 7.2.2 tells us that the maximum α for which the anytime capacity is larger than $\log_2(1.258)+4\log_2(1.058)$ is only around 0.646. This is enough for the four slow components which each require $\alpha > 2\log_2(1.058) = 0.163$. But it is less than the minimum α we require for the first component: $\alpha > 2\log_2(1.258) = 0.662$. This means that as long as we insist on treating all the bits alike in the reliable transmission layer, it is impossible to achieve a finite mean squared error on the first component

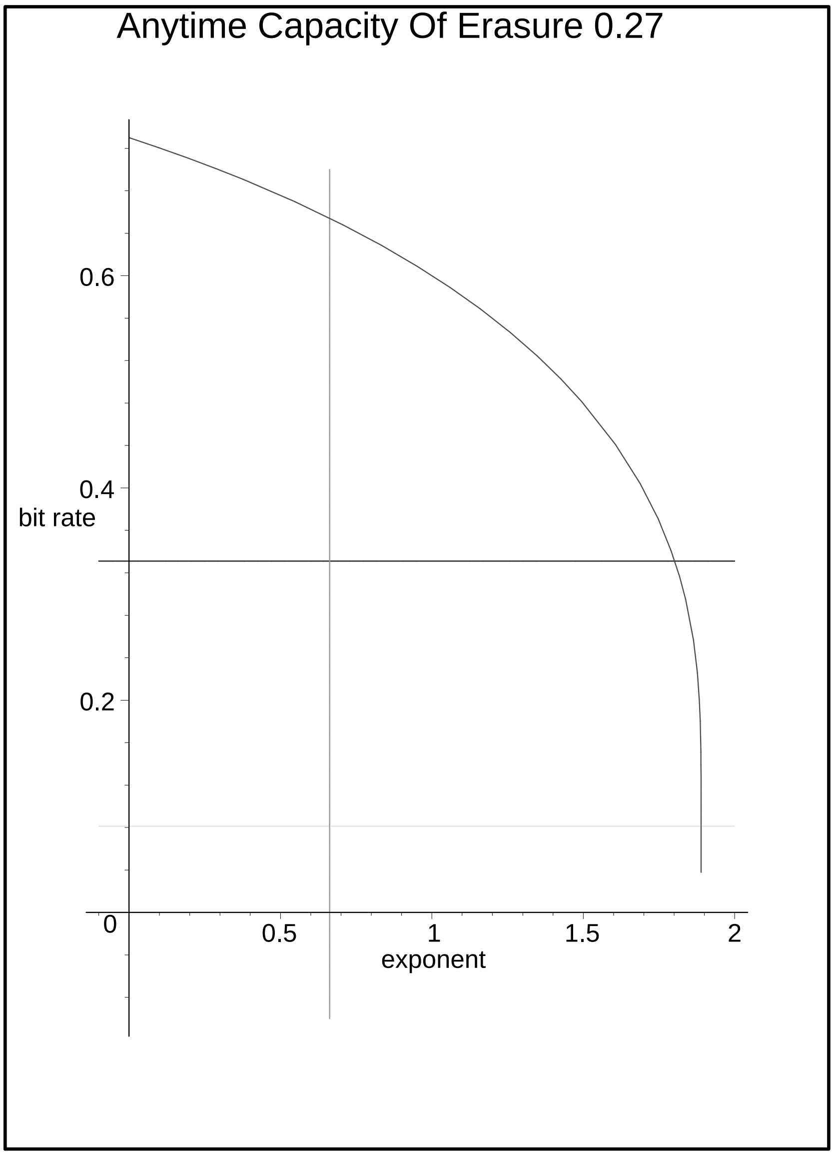

Figure 8-2: Anytime capacity for the erasure channel with $e = 0.27$ for encoders having access to noiseless feedback

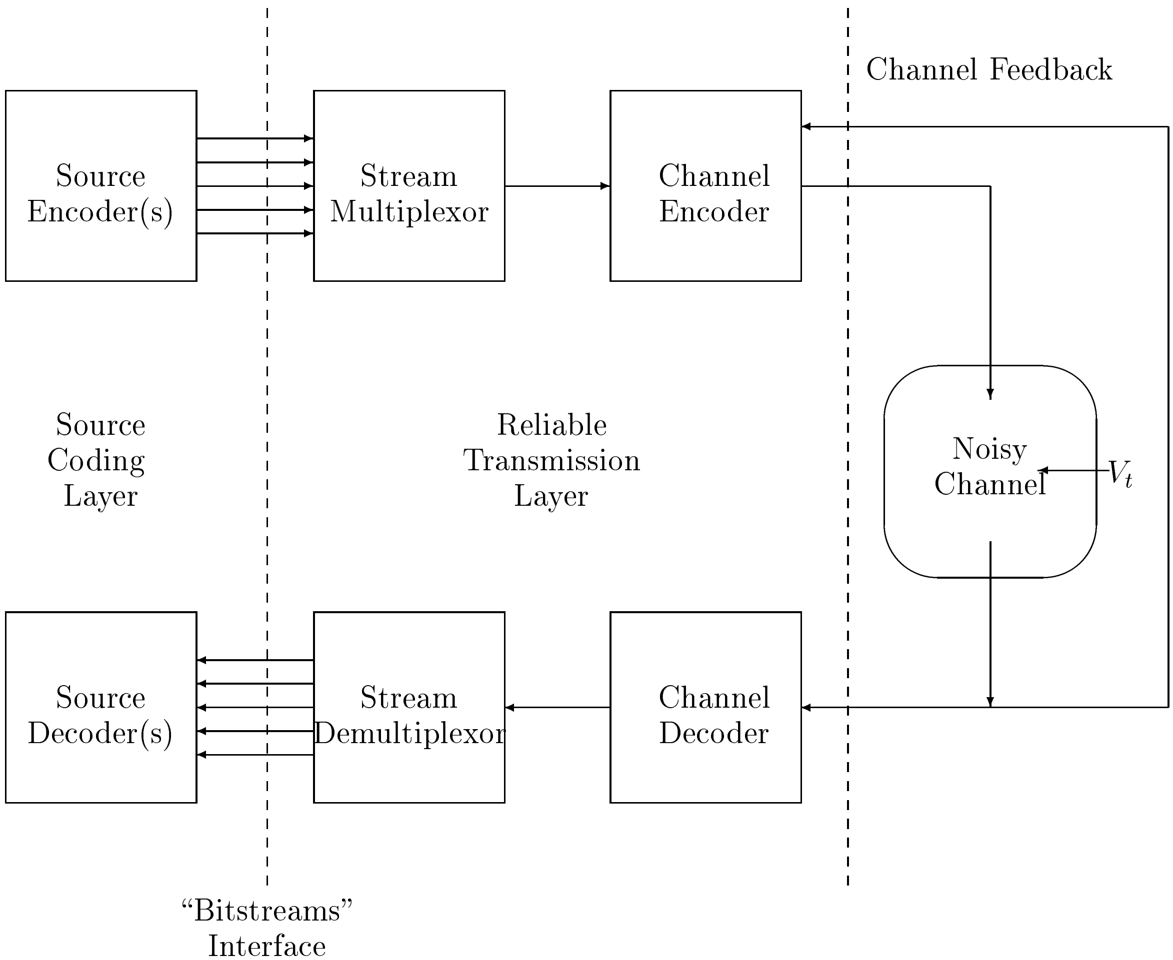

Figure 8-3: Forcing all the bitstreams to get the same treatment for reliable transmission

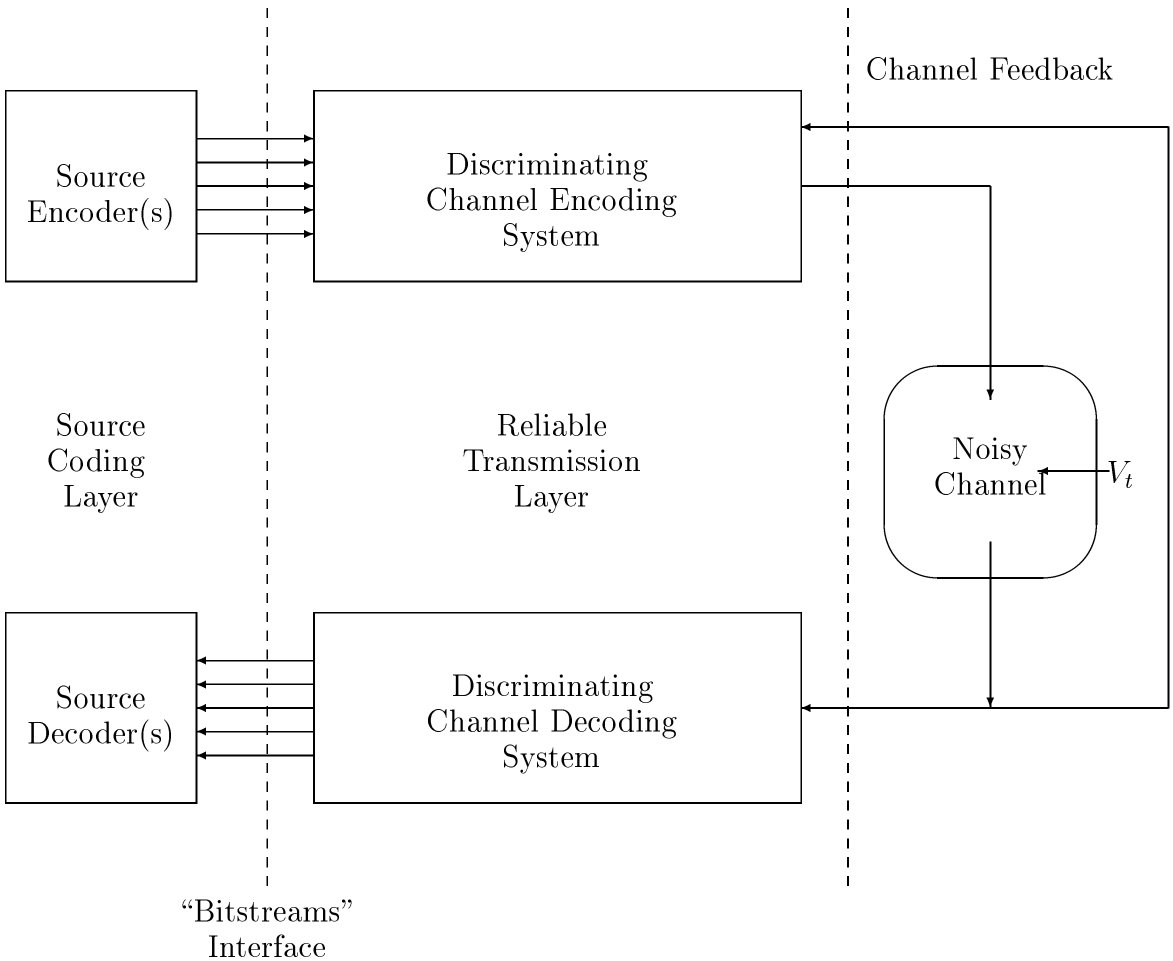

Figure 8-4: Allowing the reliable transmission layer to discriminate between bitstreams

of the transformed source. Because this is a consequence of our fundamental separation theorem for such processes, it is true regardless of how much end-to-end delay we are willing to tolerate or how we do the source encoding.

8.3 Differentiated Service

Let us consider an alternative strategy more in the spirit of "loosely coupled joint source and channel coding."[32]

For our transformed source, we can use the scalar codes from Theorem 4.1.1 to encode the first component at a rate $R_1 = \frac{1}{3} > \log_2(1.258)$. The first stream generates a bit every third time step. The other components are all encoded at a rate $R_{2,3,4,5} = \frac{1}{12} > \log_2(1.058)$ and gets the lower priority. These together generate a bit every three steps (4 bits every 12 time steps). The total rate is therefore $R = \frac{2}{3} < 1 - e = 0.73$.

As shown in Figure 8-4, we then ask the reliable transmission layer for different treatment of the bitstreams encoding different components of the transformed source.

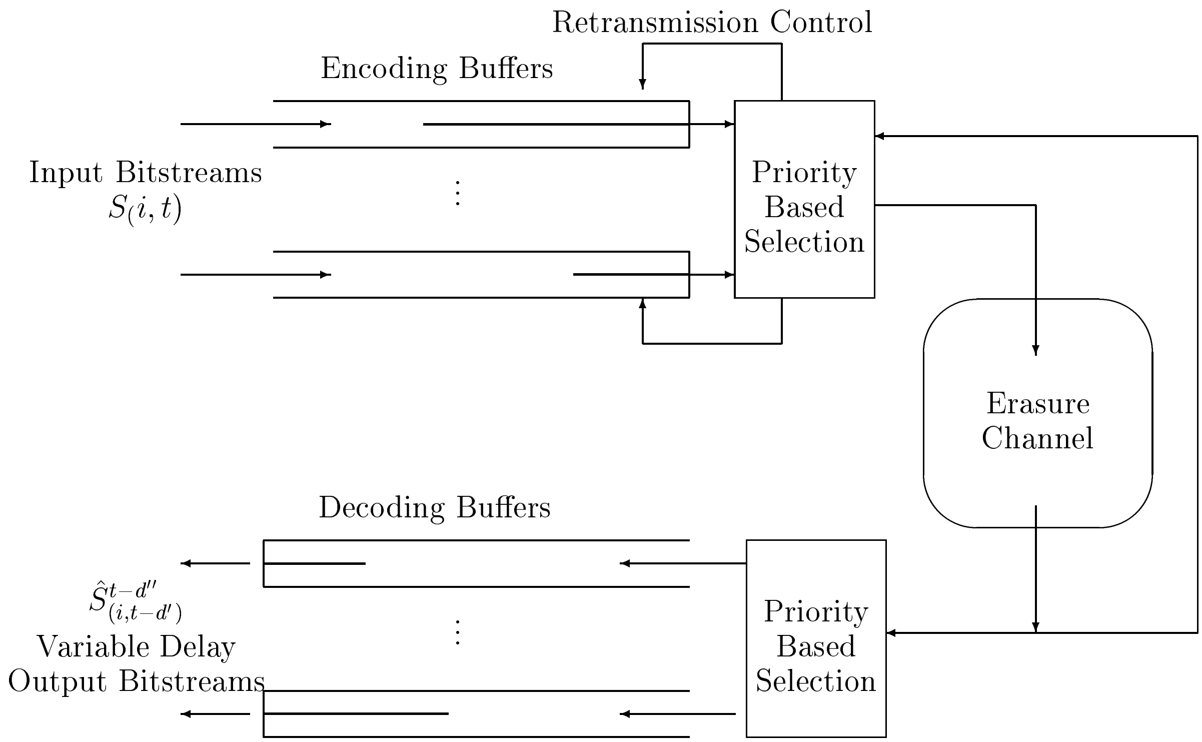

Figure 8-5: The strict priority queuing strategy for discrimination between bitstreams

8.3.1 A Prioritized Channel Code

We request a higher priority for the bitstream representing the "faster" component that requires $\alpha > 2\log_2(1.258)$. We use the strategy for reliable transmission over the erasure channel with feedback shown in Figure 8-5:

- Store the incoming bits from the different streams into prioritized FIFO buffers — one buffer for each distinct priority level.

- At every opportunity for channel use, transmit the oldest bit from the highest priority input buffer that is not empty.

- If the bit was received correctly, remove it from the appropriate input buffer.

On the channel decoding side, the anytime decoders are intuitively clear. Since there is noiseless feedback and the encoder's incoming bitstreams are deterministic in their timing, the decoder can keep track of the encoder's buffer sizes. As a result, it knows which incoming bit belongs to which stream and can pass the received bit on to the source decoding layer with an appropriate label. The source decoder takes all available bits and makes the best prediction of where it thinks the source process is.

At the receiver's side, we use all the bits we have received to make our estimates. If we are missing some bits for a given component, we just predict by multiplying what we have by the appropriate power of that component's eigenvalue.

8.3.2 Analyzing This Channel Code

The first thing we notice is that for any delay d, the channel decoder makes an error for bitstream i only if the encoder's buffer i contains more than dR_i bits still awaiting transmission. If there are fewer bits waiting, it means that the bit from d time units ago has already made it across. As a result, to analyze how the average probability of error varies with delay, we only need to study the steady-state distribution of the number of bits in each buffer awaiting transmission.

The High Priority Stream

Since the highest priority stream preempts all lower priority streams, it effectively does not have to share the channel at all. We can study its queue length using a simple Markov chain by grouping time into three time unit blocks. Then, the number of bits awaiting transmission at the end of a block is the Markov state and use $p_{i,j}$ to represent the probability that the queue in state i will go next to state j.

$$\begin{aligned}
p_{0,0} &= 3e^2(1-e) + 3e(1-e)^2 + (1-e)^3 \\
p_{i,i+1} &= e^3 \\
p_{i,i} &= 3e^2(1-e) \\
p_{i,i-1} &= \begin{cases} 3e(1-e)^2 + (1-e)^3 & \text{if } i = 1 \\ 3e(1-e)^2 & \text{if } i > 1 \end{cases} \\
p_{i,i-2} &= (1-e)^3
\end{aligned}$$

It is possible to calculate the steady state distribution π for this Markov chain. By some algebraic manipulation we can get the following recurrence relation:

$$(1-e)^3\pi_i = \begin{cases} (1-3e^2(1-e))\pi_{i-2} - 3e(1-e)^2\pi_{i-1} - e^3\pi_{i-3} & \text{if } i > 2 \\ (1-3e^2(1-e) - 3e(1-e)^2 - 3e^2(1-e))\pi_{i-2} - 3e(1-e)^2\pi_{i-1} & \text{if } i = 2 \end{cases}$$

It turns out that $\pi_i \propto \left(\frac{2e^3}{1+2e^3+(1-e)\sqrt{1+2e-3e^2}-3e^2}\right)^i$ as i gets large and thus:

$$\begin{aligned}
P_{\text{error}}(\text{Delay} = d) &\leq P(\text{Buffer State} > dR_1) \\
&\leq K\left(\frac{2e^3}{1+2e^3+(1-e)\sqrt{1+2e-3e^2}-3e^2}\right)^{\frac{d}{3}} \\
&= K2^{-\frac{1}{3}\log_2\left(\frac{1+2e^3+(1-e)\sqrt{1+2e-3e^2}-3e^2}{2e^3}\right)d}
\end{aligned}$$

which for $e = 0.27$ results in an $\alpha = 1.799 > 2\log_2(1.258) = 0.662$ so it is more than fast enough.

The Low Priority Streams

To analyze the asymptotic probability of error for the second set of streams, we notice that regardless of the realization of channel noise, the sum of the queue lengths for the two buffers is identical to the queue length for a hypothetical single stream at the combined rate. It suffices to look at a single stream with rate $\frac{2}{3}$. Once again, we will group channel uses into blocks of three so that the number of bits awaiting transmission at the end of a block is the Markov state for the system. This gives us:

$$\begin{aligned}
p_{0,0} &= 3e(1-e)^2 + (1-e)^3 \\
p_{i,i+2} &= e^3 \\
p_{i,i+1} &= 3e^2(1-e) \\
p_{i,i} &= 3e(1-e)^2 \\
p_{i,i-1} &= (1-e)^3
\end{aligned}$$

The steady state distribution π for the state can be calculated just as before. By some algebraic manipulation we get the following recurrence relation:

$$(1-e)^3\pi_i = \begin{cases} (1-3e(1-e)^2)\pi_{i-1} - 3e^2(1-e)\pi_{i-2} - e^3\pi_{i-3} & \text{if } i > 2 \\ (1-3e(1-e)^2)\pi_{i-1} - 3e^2(1-e)\pi_{i-2} & \text{if } i = 2 \\ (1-3e(1-e)^2 - 3e^2(1-e))\pi_{i-1} & \text{if } i = 1 \end{cases}$$

For large, i, we then have $\pi_i \propto \left(\frac{2e^2}{2e^2+\sqrt{4e-3e^2}-3e}\right)^i$. This rate of decay is clearly much slower than the one for the higher priority queue and thus, for large queue lengths, it dominates. Therefore we have for the bits in the lower priority queue:

$$\begin{aligned}
P_{\text{error}}(\text{Delay} = d) &\leq P\left(\text{Combined Buffer State} > d(R_2 + R_3 + R_4 + R_5)\right) \\
&\leq K\left(\frac{2e^2}{2e^2+\sqrt{4e-3e^2}-3e}\right)^{\frac{d}{3}} \\
&= K2^{-\frac{1}{3}\log_2(\frac{2e^2+\sqrt{4e-3e^2}-3e}{2e^2})d}
\end{aligned}$$

which for $e = 0.27$ results in an $\alpha = 0.285 > 2\log_2(1.058) = 0.163$ so our scheme is fast enough for all the slow components as well!

Corollary 6.2.1 then shows that the cascaded source and channel codes achieve a finite mean squared error for all components of the transformed source. Thus, the system achieves a finite mean squared error on the original source as well.

8.4 Discussion

We have presented a simple example of a vector source and a specific channel for which it is impossible to achieve finite end-to-end mean-squared-error without using some form of differentiated service at the reliable transmission layer. Furthermore, in order to evaluate the bit-pipes provided by the reliable transmission layer, we have shown that the ideas of α-anytime-capacity are relevant and have used our previous separation results to motivate a scheme which does achieve finite end-to-end performance. Similar examples could be constructed for other channels and our α parameter would continue to be useful as a fundamental way of measuring QoS for a bit-pipe.

This toy example over a single simple link is only a beginning, but it opens the door towards a real Information Theory of "Quality of Service." Even though in the real world, few processes of interest actually have asymptotically infinite variance, we conjecture that these ideas will be a useful approximation whenever the upper limit of the tolerable end-to-end delay is within the range of time for which an unstable model is applicable. We suspect that this is true for not just control problems, but many others (including multimedia ones) as well.

Chapter 9

Conclusion and Future Work

This thesis has studied the issues involved in communicating delay sensitive signals across noisy channels by focusing on the unstable scalar linear Markov processes. We established that such processes are fundamentally different from stable processes from a sensitivity point of view. On the source side, we were able to resolve a long-standing open problem regarding the infinite horizon source coding of such processes by using a new variable rate construction weaving together dithering, vector quantization, and a lossless code on the integers. On the channel side, we introduced a new sense of reliable transmission (anytime capacity) that is more demanding than the traditional Shannon sense but is weaker than the sense underlying zero-error capacity. Our keystone result is an information transmission theorem tying the source and channel sides together.

This information transmission theorem 6.2.1 and its converse theorem 6.3.1 establish that our notion of α-anytime capacity is exactly the right notion of capacity for tracking unstable processes over noisy channel. They demonstrate in a concrete mathematical setting that not all digital bits are alike. A bitstream coming from the encoding of an unstable source with one A can have more demanding "quality of service requirements" for channel transmission than one emerging from the encoding of an unstable source with a different $A' < A$. The nice part about this is that the quality of service requirements are not specified *a-priori* based on intuition or engineering arguments. They emerge as a mathematical consequence from the natural notion of distortion for such sources. They are also not tied to a specific channel.

In addition, we extended our separation theorem to control contexts by showing the necessity of anytime capacity to stabilize a scalar system over a noisy feedback link. The techniques used to prove the converse can also be used to construct anytime encoders and decoders for channels with noiseless feedback and to analyze their performance. In particular, we use them to show a double exponential convergence for anytime decoders on an AWGN channel with a power constraint. The same technique can be used to lower bound zero-error capacity for continuous channels with an average power constraint and additive noise with bounded support.

While the discussions at the end of each chapter so far point to a few of the interesting questions that this thesis directly opens up, here I will close with some more speculative future directions.

9.1 Source Encoders Without Memory

Our simple random walk source of Section 1.2 actually admits a code that works without any memory at all on the encoder side. Moreover, this code is time-invariant:

$$F'(X) = \left|\lfloor \frac{X}{2} \rfloor\right| \bmod 2 \tag{9.1}$$

Notice that since the value of X must change by 1 at each time step, that $|\lfloor \frac{X}{2} \rfloor|$ can either remain constant or change by 1 depending on the direction of motion and the starting position. Therefore, the remainder after dividing by 2 must also either change or stay the same. Since the original position is known to be the origin, intuitively the decoder should be able to follow X at each time step. Recursively, the decoder can be defined as follows:

$$G'_t(S_1^t) = G'_{t-1}(S_1^{t-1}) + (-1)^{S_t}\left(1 - 2F'(G'_{t-1}(S_1^{t-1}) + 1)\right) \tag{9.2}$$

with $G'_0 = 0$ as the base case. However, this decoder is more sensitive to errors than the previous one. Each error not only contributes 1 unit of deviation, it also reverses the sense in which future bits are perceived, further compounding the error with time.

It turns out that the intrinsic sensitivity is the same as for the code without memory:

Proposition 9.1.1 *For the rate* 1 *code given by equations (9.1) and (9.2),* $\Delta^-(d) = \infty$ *for all* d *while* $(d+1)^2 \leq \Delta^+(d) \leq 4((d+1))^2$ *if* $d \geq 0$ *and zero otherwise.*

Proof: Essentially identical to Proposition 3.3.1. The only difference is in the worst case set of bit errors. □

Even if we use the memoryless source code F' from (9.1) with an anytime encoder for the channel, the combined encoder as a whole will have memory. However, the existence of F' is suggestive and we are currently looking into whether it is possible to track this simple source using a memoryless joint-source-channel encoder. We suspect that it is possible, though perhaps at the cost of some additional rate.

The general issue of how much memory is needed and where it should be placed in a complete system is an interesting one that needs to be studied.

9.2 Sequential Refinement For Source Codes

Shannon himself commented on the interesting duality between the problems of source and channel coding. Our introduction of anytime codes for reliable transmission of bits across noisy channels prompts a speculative question: what is the analogous concept for source coding? Imagine that our source generated i.i.d. bits equally likely to be 0 or 1. Use standard Hamming distortion as the distortion measure. An anytime code can be viewed as a joint source/channel code for this source and distortion measure which manages to improve the reconstructions of the original source as time goes on while simultaneously allowing for estimates of recent source symbols previously unestimated at the decoder. Its "delay universality" can also be viewed as achieving the rate-distortion bounds over the noisy channel without having to specify a delay in advance.

It is natural to wonder whether general source codes exist which have a similar property. Do there exist source encoders which map a source stream into a bitstream such that we can get arbitrarily close to the rate-distortion limits on distortion purely by adjusting the delay on the source decoding? The codes constructed in Chapter 4 of this thesis do not have this property. But it would be fascinating to find out if such codes exist or to be able to prove that they do not.

9.3 Distributed Systems and Multiterminal Information Theory

The simple toy control systems of chapter 7 are only the beginning. The ultimate goal should be getting a handle on communication within general distributed systems. In such systems the transmission of information cannot be divorced from its use. A complete understanding of concepts like QoS requirements in networks requires us to be able to deal with transmission and use together in an intelligent way. We hope that the basic ideas of splitting and combining channels will be refined and extended onto the source side as well to cover the splitting and combining of sources into bitstreams. Such a unified theory may be a way to get a firm grasp on the elusive concept of "meaning" in systems.

Appendix A

Streaming Definitions

A lot of information theory is traditionally described in the "one-shot setting" where time and the order of events is not a big concern. However, for this thesis, time and the order of events is very important since we are concerned with "information streams" where random variables are only realized at particular times and are not available beforehand. We are also interested in closed-loop systems where causality is critical. Without some form of causality, a model with feedback ceases to be intelligible. In this appendix, we give streaming definitions of basic objects like sources, channels, encoders, decoders, etc. that emphasize the times at which everything becomes available and the resulting causal dependencies between variables.

A.1 Streams and Transition Maps

Definition A.1.1 *Given a real number $\tau > 0$ and real number θ, a $(\theta, \tau, \mathcal{A})$* stream *$A$ is a sequence of variables A_1^∞ such that $A_i \in \mathcal{A}$ and A_i occurs at time $\tau(\theta + i)$.*

τ represents the time between samples and the offset θ is normalized so that -1 shifts samples up by one and $+1$ delays them by one regardless of the inter-sample time τ.

Definition A.1.2 *Given two positive real numbers τ_1, τ_2, two real numbers θ_1, θ_2, and two alphabets $\mathcal{A}, \mathcal{B}$, a (θ_1, θ_2)* offset *$(\tau_1, \tau_2, \mathcal{A}, \mathcal{B})$* transition-map *$\mathcal{E}$ is a sequence of functions $\mathcal{E}_i$ from $\mathcal{A}^{\lfloor \frac{(\theta_2+i)\tau_2}{\tau_1} - \theta_1 \rfloor}$ into $\mathcal{B}$.*

It is clear that a transition-map causally transforms one stream into another: the output at time $\tau_2(\theta_2 + j)$ does not depend on a_i that occur after that time. Some transition maps have even less of dependence on the past.

Definition A.1.3 *A (θ_1, θ_2) offset $(\tau_1, \tau_2, \mathcal{A}, \mathcal{B})$ transition-map $\mathcal{E}$ is* memoryless with respect to its input *if $\mathcal{E}_i$ depends only on the $a_{(\lfloor \frac{(\theta_2+(i-1))\tau_2}{\tau_1} - \theta_1 \rfloor + 1)}^{\lfloor \frac{(\theta_2+i)\tau_2}{\tau_1} - \theta_1 \rfloor}$ and not any of other past values of the input.*

Sometimes, we want to allow the possibility of an external input.

Definition A.1.4 *Similarly, a $\mathcal{V}$* dependent *(θ_1, θ_2) offset $(\tau_1, \tau_2, \mathcal{A}, \mathcal{B})$ transition-map $\mathcal{E}^V$ is one in which the functions $\mathcal{E}_i$ have a further dependence on $V \in \mathcal{V}$. So $\mathcal{E}_i$ maps $\mathcal{A}^{\lfloor \frac{(\theta_2+i)\tau_2}{\tau_1} - \theta_1 \rfloor} \times \mathcal{V}$ into $\mathcal{B}$.*

As stated above, V is generally realized all at once and is the same for the entire sequence of functions above. However, this need not be the case. In many interesting cases, $\mathcal{V} = \mathcal{V}_0 \times \check{\mathcal{V}}^\infty$ and the $\check{V}_1^\infty \in \check{\mathcal{V}}^\infty$ variables are realized sequentially at regular intervals in time. Then causality matters and we have:

Definition A.1.5 *A* $(\theta', \tau', \mathcal{V}' = \mathcal{V}_0 \times \check{\mathcal{V}}^\infty)$ causally dependent (θ_1, θ_2) *offset* $(\tau_1, \tau_2, \mathcal{A}, \mathcal{B})$ *transition-map* $\mathcal{E}^V$ *is one in which the functions* $\mathcal{E}_i$ *map* $\mathcal{A}^{\lfloor \frac{(\theta_2+i)\tau_2}{\tau_1} - \theta_1 \rfloor} \times \mathcal{V}_0 \times \check{\mathcal{V}}^{\lfloor \frac{(\theta_2+i)\tau_2}{\tau'} - \theta' \rfloor}$ *into* $\mathcal{B}$.

Such a causal dependency condition could equivalently be written as a restriction that the functions $\mathcal{E}_i$ do not depend on the parts of V that occur in the future. An analogous notion of memoryless dependence also exists. Furthermore, we can clearly string together $(\theta', \tau', \mathcal{V}')$ causal (or memoryless) dependency clauses to have the output depend on many different incoming streams.

A.1.1 Pairings

We would like to be able to interconnect such transition-maps. There is a natural concatenation operation corresponding to connecting output to input:

Definition A.1.6 *A* (θ_1, θ_2) *offset* $(\tau_1, \tau_2, \mathcal{A}, \mathcal{B})$ *transition map* $\mathcal{E}'$ *and* (θ_3, θ_4) *offset* $(\tau_3, \tau_4, \mathcal{C}, \mathcal{D})$ *transition map* $\mathcal{E}''$ *can be combined into a single* (θ_1, θ_4) *offset* $(\tau_1, \tau_4, \mathcal{A}, \mathcal{D})$ *transition map* $\mathcal{E} = \mathcal{E}' \circ \mathcal{E}''$ *if* $\theta_2 = \theta_3$, $\tau_2 = \tau_3$, *and* $\mathcal{B} = \mathcal{C}$. *The functions are given by:*

$$\mathcal{E}_i(a_1^{\lfloor \frac{(\theta_4+i)\tau_4}{\tau_1} - \theta_1 \rfloor}) = \mathcal{E}_i''(\mathcal{E}_1'(a_1^{\lfloor \frac{(\theta_2+1)\tau_2}{\tau_1} - \theta_1 \rfloor}), \mathcal{E}_1'(a_1^{\lfloor \frac{(\theta_2+2)\tau_2}{\tau_1} - \theta_1 \rfloor}), \ldots, \mathcal{E}_1'(a_1^{\left\lfloor \frac{(\theta_2 + \lfloor \frac{(\theta_4+i)\tau_4}{\tau_3} - \theta_3 \rfloor)\tau_2}{\tau_1} - \theta_1 \right\rfloor}))$$

It is similarly possible to connect a $(\theta, \tau, \mathcal{B})$ output stream from one transition map to any transition map that is $(\theta, \tau, \mathcal{V}_0 \times \mathcal{B}^\infty)$ causally dependent. In any sort of concatenation, we must take some care to avoid "causality loops" in which a variable at a given time ends up depending on its own value at the same time.

There is another kind of conceptual pairing possible. In many cases, we want to think of these maps as occurring in encoder/decoder pairs in which the input to the encoder and the output of the decoder are in a natural one-to-one correspondence though they need not be physically connected to each other.

Definition A.1.7 *An* encoder/decoder pair *is a pair* (F, G) *of transition maps together with a* reconstruction profile r_1^∞ *of positive integers satisfying the following properties.*

- *If* F *is a* (θ_1, θ_2) *offset* $(\tau_1, \tau_2, \mathcal{A}, \mathcal{B})$ *transition-map, then* G *is a* (θ_3, θ_4) *offset* $(\tau_2, \tau_1, \mathcal{C}, \mathcal{D})$ *transition-map.*
- *The sequence* r_1^∞ *contains no repetitions.*

The reconstruction *of input* a_i *is* $d_{r_i} = G_{r_i}(c_1^{\lfloor \frac{(\theta_4+r_i)\tau_1}{\tau_2} - \theta_3 \rfloor})$.

The delay *of the encoder/decoder pair is* $\sup_{i \geq 1}(r_i - i + \theta_1 - \theta_4)\tau_1$. *We only accept reconstruction profiles* r_1^∞ *that are* non-anticipatory*: the minimum delay* $\inf_{i \geq 1}\{(r_i - i + \theta_1 - \theta_4)\tau_1\} \geq 0$

A.1.2 Stream Constraints

We also need a notion of an acceptable constraint on a stream.

Definition A.1.8 *A* stream constraint *on a $(\theta, \tau, \mathcal{A})$ stream is a statement or sequence of statements regarding the variables in the stream.*

A sequential constraint *F is a sequence of functions F_i mapping $\mathcal{A}^i$ into $\Re$ together with the statement "$\forall i > 0, F_i(A_1^i) \leq 0$"*

A limiting constraint *F is a sequence of functions F_i mapping $\mathcal{A}^i$ into $\Re$ together with the statement "$\limsup_{i\to\infty} F_i(A_1^i) \leq 0$"*

In a completely specified probabilistic system, a stream constraint is met *if the event making the statement true has probability* 1.

In general, we will try to make the underlying space of primitive random variables be an i.i.d. collection of uniform random variables on the unit interval.

A.2 Random Sources

A source is a random process evolving in discrete time. To normalize the units of time, we assume that the source generates one symbol $X_t \in \mathcal{X}$ at every unit time. Furthermore, to avoid confusion, every source we will consider will be the output of a transition map driven by a stream of i.i.d. uniform random variables on the real interval $[0, 1]$.

Definition A.2.1 *A* random source *is a* $(0, 0)$ *offset* $(1, 1, [0, 1], \mathcal{X})$ *transition-map X. For convenience, X_t is also used to denote the random variable that results after connecting the input of the transition map to a* $(0, 1, [0, 1])$ *stream Q of independent real-valued random variables uniform on the interval* $[0, 1]$.

This definition is adopted so that our formulation will always have a simple set of primitive random variables.

A.2.1 Markov Sources

Our main focus is on scalar valued Markov processes.

Definition A.2.2 *Given a real number a, and functions W_t from $[0, 1]$ to $\Re$, the* scalar discrete-time Markov source with parameter A and noise W *is defined by:*

$$\begin{aligned} X_0 &= 0 \\ X_t &= AX_{t-1} + W_t(Q_t) \end{aligned}$$

This can be expressed without recursion as:

$$X_t(Q_1^t) = \sum_{i=1}^{t} A^{t-i} W_i(Q_i)$$

We will usually assume that the functions $W_t = W_0$ and that the range of W_0 is bounded.

A.3 Source Codes

We are now ready to define source codes. Given a source alphabet $\mathcal{X}$ and reconstruction alphabet $\hat{\mathcal{X}}$ and a discrete-time source $\{X_t\}$ which generates a new symbol every unit time:

Definition A.3.1 *A* rate R source code *is an encoder/decoder pair* (F, G) *such that* F *is a* $(0,0)$ *offset* $(1, \frac{1}{R}, \mathcal{X}, \{0,1\})$ *transition map and* G *is a* $(0,0)$ *offset* $(\frac{1}{R}, 1, \{0,1\}, \hat{\mathcal{X}})$ *transition map.*

It should be clear how ((1.1), (1.2)) and ((9.1), (9.2)) are both rate 1 source codes with zero delay.

We defined source code in such a way as to map any source into a binary representation and back again to the reconstruction alphabet. This allows us to concatenate the encoder/decoder pair together without having to put anything else in between.

A.4 Noisy Channels

We can consider noisy channels as transition maps. While sources were transition maps with an i.i.d. input stream, noisy channels are transition maps causally dependent on an i.i.d. stream.

Definition A.4.1 *Let* V_i *be independent and identically distributed uniform random variables on the real interval* $[0,1]$ *and let* $\mathcal{V} = [0,1]^\infty$*. A* discrete-time noisy channel *is a* $(\theta_2, \tau, \mathcal{V})$ *causally dependent* (θ_1, θ_2) *offset* $(\tau, \tau, \mathcal{A}, \mathcal{B})$ *transition-map* $\mathcal{T}^V$*. The* transmission delay *is* $(\theta_1 - \theta_2)\tau$.

A memoryless time-invariant discrete-time channel *is a discrete-time noisy channel for which there exists a function* $\mathcal{T}_0$ *such that* $\mathcal{T}_i^V(a_1^{i+\lfloor\theta_2-\theta_1\rfloor}) = \mathcal{T}_0^{V_i}({a'}_{i+\lfloor\theta_2-\theta_1\rfloor})$ *for all* i.

We will focus on memoryless time-invariant discrete-time channels with zero transmission delay ($\theta_1 = \theta_2 = 0$) and inter-symbol time τ. Sometimes, we will also couple the noisy channel with a stream constraint on the input to the channel. Our formulation of channels as transition maps with an explicit causal dependence on an i.i.d. stream looks superficially different from the standard definitions[11] which tend to be given in terms of transition probabilities or constraints. To see that it includes the standard channels we think about, we now give definitions of them in our terms.

A.4.1 Example Channels

The simplest noisy channel is the following:

Definition A.4.2 *The* binary erasure channel *is a memoryless time-invariant discrete-time channel with* $\mathcal{A} = \{0,1\}$, $\mathcal{B} = \{0,1,\emptyset\}$ *and*

$$\mathcal{T}_0^V(a) = \begin{cases} \emptyset & \text{if} V \le e \\ a & \text{otherwise} \end{cases}$$

As we can see, the binary erasure channel has probability of erasure e and otherwise it transmits the incoming bit without error. Expressed in terms of transition probabilities, $p(0|0) = p(1|1) = 1 - e$ while $p(\emptyset|0) = p(\emptyset|1) = e$.

Another important class of channels are the additive noise channels. These are routinely used to model continuous valued channels. The Additive White Gaussian Noise (AWGN) is the most popular:

Definition A.4.3 *A* scalar AWGN channel with variance K_V and power constraint P *is a memoryless time-invariant discrete-time channel with* $\mathcal{A} = \mathcal{B} = \Re$ *and* $\mathcal{T}_0^V(a) = a + N^{-1}(V)$ *where* $N(x) = \frac{1}{\sqrt{2\pi K_V}} \int_0^x e^{-\frac{s^2}{2K_V}} ds$ *together with the following limiting stream constraint on the input stream:* $\limsup_{i \to \infty} \frac{\sum_{j=1}^{i} a_j^2}{i} - P \leq 0$

The power constraint is needed to prevent degenerate solutions. We have stated it as a limiting constraint while it is usually stated as a constraint on the expectation of the power. If every random variable in the system were ergodic and stationary, the two kinds of statements are practically interchangeable. But because we want to allow non-ergodic and non-stationary processes to exist within our models, we use this limiting form to make it clear what we mean.

A.5 Channel Codes

A channel code is an encoder/decoder pair that can be wrapped around a noisy channel.

Definition A.5.1 *Assume that the noisy channel has inter-sample time* τ *and offsets* (θ_1, θ_2). *A* θ offset, rate R channel code *is an encoder/decoder pair* $(\mathcal{E}, \mathcal{D})$ *such that channel encoder* $\mathcal{E}$ *is a* (θ, θ_1) *offset* $(\frac{1}{R}, \tau, \{0,1\}, \mathcal{A})$ *transition map and channel decoder* $\mathcal{D}$ *is a* (θ_2, θ) *offset* $(\tau, \frac{1}{R}, \mathcal{B}, \{0,1\})$ *transition map.*

The rate R in the definition is given in bits per unit time. It can be multiplied by τ to give the rate in bits per channel use. For convenience, we will denote the bits coming into the encoder as S_1^∞ and their corresponding reconstructions as $\hat{S}_1^\infty$. Bit S_i is received at time $\frac{\theta+i}{R}$ while the reconstruction $\hat{S}_i$ happens at time $\frac{\theta+r_i}{R}$.

The whole idea of a channel code is to reconstruct the input bits reliably at the output of the decoder.

A.5.1 Block Channel Codes

The definitions we have used are a bit unusual. To justify them, we will show how they include the standard notion of block-codes as a special case.

Definition A.5.2 *For non-negative integers* R_{in} *and* R_{out}, *a* (R_{in}, R_{out}) block channel encoder *is a function* $\mathcal{E}_0$ *from* $\{0,1\}^{R_{in}}$ *into* $\mathcal{A}^{R_{out}}$. *Similarly, a* (R_{in}, R_{out}) block channel decoder *is a function* $\mathcal{D}_0$ *from* $\mathcal{B}^{R_{out}}$ *into* $\{0,1\}^{R_{in}}$. *The block code has rate* $\frac{R_{in}}{R_{out}}$ *bits per channel use or* $\frac{R_{in}}{\tau R_{out}}$ *bits per unit time.*

In the case where the original bits are generated one at a time in a regular stream, the block encoder works as follows:

1. Buffer incoming bits until we have R_{in} of them

2. Apply $\mathcal{E}_0$ to get a codeword to send over the channel.

3. Send out the codeword one symbol at a time while doing 1 and preparing for the next block.

And similarly, the block decoder does this:

1. Buffer channel outputs until we have R_{out} of them
2. Apply $\mathcal{D}_0$ to find out which codeword and hence which block of input bits was sent.
3. Send out the decoded bits one at a time while doing 1 and preparing for the next block.

To calculate the delay of a block channel code, we notice that the encoder must wait $(R_{in} - 1)\frac{\tau R_{out}}{R_{in}}$ units of time in buffering before the relevant codeword can even begin to be transmitted. Similarly, the decoder must wait another $(R_{out} - 1)\tau$ while the codeword is being received. This gives us a minimum delay of at least $\tau(2R_{out} - 1 - \frac{R_{out}}{R_{in}})$ units of time. Aside from the effect of any offsets, this is the delay experienced by the first bit in the block even though the block decoder could potentially reconstruct the original codeword (and hence all the R_{in} bits that went into choosing it) as soon as the last part of it is received across the channel. This gives us the following:

Lemma A.5.1 *For any (R_{in}, R_{out}) block channel code $(\mathcal{E}_0, \mathcal{D}_1)$, there exists a rate $R = \frac{R_{in}}{\tau R_{out}}$ channel encoder $\mathcal{E}$, decoder $\mathcal{D}$, and reconstruction profile r_1^∞ that generates the same sequence of $\hat{s}_1^\infty$ given the same source sequence s_1^∞ and channel noise. Moreover, the reconstruction profile has delay $\tau(2R_{out} - 1 - \frac{R_{out}}{R_{in}})$ units of time.*

A.5.2 Codes with access to feedback

It is sometimes useful to consider situations in which the decoder has some medium to talk back to the encoder. In our view, we can do this by allowing the channel encoder to have an additional causal dependence on the relevant feedback stream. In general, this stream might also be noisy or constrained in some way. However, we will restrict our attention to the case of noiseless feedback from the output of the channel back to the encoder.

Definition A.5.3 *Assume that the noisy channel has inter-sample time τ and offsets (θ_1, θ_2). A θ* offset, rate R channel encoder with access to noiseless feedback *$\mathcal{E}$ is a (θ, θ_1) offset $(\frac{1}{R}, \tau, \{0,1\}, \mathcal{A})$ transition map $(\theta_2 + \delta, \tau, \mathcal{B}^\infty)$ causally dependent on the output stream of the channel. $\delta > 0$ is chosen so that $(\theta_2 + \delta) > \theta_1$.*

The additional delay δ is chosen to prevent problems with causality loops — ensuring that even if the channel has no intrinsic delay, the encoder at a given time only has access to the strictly prior channel outputs.

Appendix B

Convergence To Means

B.1 Chebychev inequality

The standard Chebychev inequality is a well known result that bounds the probability of a large deviation from the mean for any distribution with a finite variance.

Theorem B.1.1 *Given i.i.d. random variables $\{X_i\}$ with common finite mean $\bar{X}$ and finite variance σ^2, the empirical average $\frac{1}{N}\sum_{i=1}^{N} X_i$ converges to $\bar{X}$ as follows:*

$$P(|\frac{1}{N}\sum_{i=1}^{N} X_i - \bar{X}| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2 N}$$

Proof: Although this is a well known result, it is illustrative to look at the simple proof to see why it cannot easily be generalized to fractional moments.

$$\begin{aligned}
E\left[(\frac{1}{N}\sum_{i=1}^{N} X_i - \bar{X})^2\right] &= \frac{1}{N^2} E\left[(\sum_{i=1}^{N} X_i - \bar{X})^2\right] \\
&= \frac{1}{N^2} E\left[\sum_{i=1}^{N}(X_i - \bar{X})^2 + \sum_{i=1}^{N}\sum_{j\neq i}(X_i - \bar{X})(X_j - \bar{X})\right] \\
&= \frac{1}{N^2}(\sum_{i=1}^{N} E\left[(X_i - \bar{X})^2\right] + \sum_{i=1}^{N}\sum_{j\neq i} E\left[(X_i - \bar{X})(X_j - \bar{X})\right]) \\
&= \frac{1}{N^2}(\sum_{i=1}^{N} \sigma^2 + \sum_{i=1}^{N}\sum_{j\neq i} E\left[(X_i - \bar{X})\right] E\left[(X_j - \bar{X})\right]) \\
&= \frac{\sigma^2}{N}
\end{aligned}$$

There are two key steps in the above derivation. The first is that we are able to express the square of the sum as a sum of products and then bring the expectation inside the sum. The other key step is where all the cross correlation terms go to zero since the expectation of a product of two independent random variables is the product of expectations. Then, we can just apply Markov's inequality to get:

$$P(|\frac{1}{N}\sum_{i=1}^{N} X_i - \bar{X}| \geq \epsilon) = P((\frac{1}{N}\sum_{i=1}^{N} X_i - \bar{X})^2 \geq \epsilon^2)$$

$$\begin{aligned} &\leq \frac{E\left[(\frac{1}{N}\sum_{i=1}^{N} X_i - \bar{X})^2\right]}{\epsilon^2} \\ &= \frac{\sigma^2}{\epsilon^2 N} \end{aligned}$$

All the above works just as well with only pairwise independence. □

If all positive integer moments exist, and the integral defining the moment generating function $\int e^{sx} dF(x)$ converges to finite values for some positive values of s, then the same basic approach can also give rise to the faster exponential convergence bound known as the Chernoff Bound.

B.2 Distributions with Heavy Tails

In some cases, the variance is infinite. The above approach does not work if all we have is a finite fractional moment $1 < \beta < 2$. This is because the fractional power of a sum cannot easily be expressed as a sum of products. A power-series expansion could be used, but it would only converge in a small neighborhood and thus would not let us bring the expectation inside the sum.

However, an alternative approach[68, 8, 44] does succeed and gives us:

Theorem B.2.1 *Given i.i.d. random variables $\{X_i\}$ with common finite mean $\bar{X}$ and finite β-th moment $E[|X_i|^\beta] < \infty$ for $\beta \in (1,2)$, the empirical average $\frac{1}{N}\sum_{i=1}^{N} X_i$ converges to $\bar{X}$ as follows: $\forall \alpha \in (1, \beta)$ there exists a distribution dependent constant K such that:*

$$P(|\frac{1}{N}\sum_{i=1}^{N} X_i - \bar{X}| \geq \epsilon) \leq \frac{K}{\epsilon^\alpha} N^{1-\alpha}$$

Proof: Since the result is not well known and the literature does not appear to have a single place where the result is derived from only elementary facts, I give a complete proof below. I follow [68, 8, 44] relatively closely. The proof works by looking at the Fourier transform (characteristic function) of the distribution for $Y_i = (X_i - \bar{X})$. Let $F(y) = P(Y_i \leq y)$ and consider:

$$f(s) = \int_{-\infty}^{\infty} e^{isY} dF(y)$$

for real values of s. It is clear that $f(s) = 1 - \Psi(s)$ (with $\Psi(0) = 0$) and it turns out that the probability of large deviations is related to the behavior of the real part of $\Psi(s)$ near the point $s = 0$. We show that $\Psi(s)$ is $O(s^\alpha)$ near the origin. First, we take a look at the real part:

$$\begin{aligned} \Re(\Psi(s)) &= s\int_0^{\frac{1}{s}} \Re(\Psi(s))dv \\ &= s \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \frac{\Re(\Psi(s))}{s} \\ &= s \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \frac{\Re(\Psi(\frac{si}{2N} + \frac{s(2N-i)}{2N}))}{\frac{si}{2N} + \frac{s(2N-i)}{2N}} \\ &\leq s \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \frac{\Re(\Psi(\frac{si}{2N}))}{\frac{si}{2N}} + \frac{\Re(\Psi(\frac{s(2N-i)}{2N}))}{\frac{s(2N-i)}{2N}} \end{aligned}$$

$$
\begin{aligned}
&= 2\lim_{N\to\infty}\frac{s}{2N}\sum_{i=0}^{2N}\frac{\Re(\Psi(\frac{si}{2N}))}{\frac{si}{2N}} \\
&= 2\int_0^s \frac{\Re(\Psi(v))}{v}dv
\end{aligned}
$$

The inequality in the above sequence comes from the fact that $\frac{\Re(\Psi(x+y))}{x+y} \leq \frac{\Re(\Psi(x))}{x} + \frac{\Re(\Psi(y))}{y}$. To see this, consider real values $s_1, s_2, s_3, z_1, z_2, z_3$ such that $(z_1 + z_2 + z_3) = 0$.

$$
\begin{aligned}
\sum_{k=1}^{3}\sum_{j>k}(\Psi(s_j - s_k)z_j z_k + \Psi(s_k - s_j)z_k z_j) &= \sum_{k=1}^{3}\sum_{j=1}^{3}\Psi(s_j - s_k)z_j z_k \\
&= \sum_{k=1}^{3}\sum_{j=1}^{3} z_j z_k - \sum_{k=1}^{3}\sum_{j=1}^{3} f(s_j - s_k)z_j z_k \\
&= (\sum_{k=1}^{3} z_j)^2 - \sum_{k=1}^{3}\sum_{j=1}^{3} f(s_j - s_k)z_j z_k \\
&= -\sum_{k=1}^{3}\sum_{j=1}^{3} f(s_j - s_k)z_j z_k \\
&= -\int_{-\infty}^{\infty}\sum_{j=1}^{3}\sum_{k=1}^{3} e^{i(s_j - s_k)u} z_j z_k dF(u) \\
&= -\int_{-\infty}^{\infty}(\sum_{j=1}^{3} e^{is_j u} z_j)(\sum_{k=1}^{3} e^{-is_k u} z_k)dF(u) \\
&= -\int_{-\infty}^{\infty}|(\sum_{j=1}^{3} e^{is_j u} z_j)|^2 dF(u) \\
&\leq 0
\end{aligned}
$$

Consider real $x, y > 0$ and let $s_1 = x+1$, $s_2 = 1-y$, $s_3 = 1$ and $z_1 = y$, $z_2 = x$, $z_3 = -(x+y)$. It is easy to see that

$$
\begin{aligned}
0 &\geq \sum_{k=1}^{3}\sum_{j>k}(\Psi(s_j - s_k)z_j z_k + \Psi(s_k - s_j)z_k z_j) \\
&= (\Psi((x+1) - (1-y))yx + \Psi((1-y) - (x+1))xy) \\
&\quad +(\Psi((x+1) - 1)y(-(x+y)) + \Psi(1 - (x+1))(-(x+y))y) \\
&\quad +(\Psi((1 - (1-y))(-(x+y))x + \Psi((1-y) - 1)x(-(x+y))) \\
&= (\Psi(x+y)xy + \Psi(-(x+y))yx) + (\Psi(x)(-(x+y))y + \Psi(-x)y(-(x+y))) \\
&\quad +(\Psi(y)(-(x+y))x + \Psi(-y)x(-(x+y)))
\end{aligned}
$$

which implies

$$
\frac{\Psi(x+y) + \Psi(-(x+y))}{x+y} \leq \frac{\Psi(x) + \Psi(-x)}{x} + \frac{\Psi(y) + \Psi(-y)}{y}
$$

which in turn gives

$$\frac{\Re(\Psi(x+y))}{x+y} \leq \frac{\Re(\Psi(x))}{x} + \frac{\Re(\Psi(y))}{y}$$

Now, we look at $\frac{\Re(\Psi(z))}{z^\alpha}$:

$$\begin{aligned}
\frac{\Re(\Psi(z))}{z^\alpha} &\leq 2z^{-\alpha}\int_0^z \frac{\Re(\Psi(s))}{s}ds \\
&= 2z^{-\alpha}\int_0^z\int_{-\infty}^{\infty}\frac{1-\cos sy}{s}dF(y)ds \\
&= 2z^{-\alpha}\int_{-\infty}^{\infty}\int_0^z\frac{1-\cos sy}{s}dsdF(y) \\
&= 2z^{-\alpha}\int_{-\infty}^{\infty}\int_0^{zy}\frac{1-\cos v}{v}dvdF(y) \\
&= 2z^{-\alpha}(\int_{-\infty}^{0}\int_0^{zy}\frac{1-\cos v}{v}dvdF(y) + \int_0^\infty\int_0^{zy}\frac{1-\cos v}{v}dvdF(y)) \\
&= 2z^{-\alpha}(\int_0^{-\infty}\int_0^{-zy}\frac{1-\cos v}{v}dvdF(y) + \int_0^\infty\int_0^{zy}\frac{1-\cos v}{v}dvdF(y))
\end{aligned}$$

The double integrals are each over a triangular area in the (y,v) plane and these areas can be rewritten to get:

$$\begin{aligned}
\frac{\Re(\Psi(z))}{z^\alpha} &\leq 2z^{-\alpha}(\int_0^\infty\int_{-\frac{v}{z}}^{-\infty}\frac{1-\cos v}{v}dF(y)dv + \int_0^\infty\int_{\frac{v}{z}}^{\infty}\frac{1-\cos v}{v}dF(y)dv) \\
&= 2z^{-\alpha}\int_0^\infty\frac{1-\cos v}{v}(\int_{-\frac{v}{z}}^{-\infty}dF(y) + \int_{\frac{v}{z}}^{\infty}dF(y))dv \\
&= 2\int_0^\infty\frac{1-\cos v}{v^{1+\alpha}}\frac{v^\alpha}{z^\alpha}P(|Y|\geq\frac{v}{z})dv \\
&\leq 2\int_0^\infty\frac{1-\cos v}{v^{1+\alpha}}\frac{v^\alpha}{z^\alpha}\max\left\{1,\frac{E[|Y|^\beta]}{(\frac{v}{z})^\beta}\right\}dv
\end{aligned}$$

Where the inequality is due to a simple application of Markov's inequality to $|Y|^\beta$. Next, we divide the integral into three parts and bound them separately. Pick a γ such that $\gamma z < \frac{1}{2}$.

$$\begin{aligned}
\frac{\Re(\Psi(z))}{z^\alpha} &\leq 2(\int_0^{\gamma z}\frac{1-\cos v}{v^{1+\alpha}}\frac{v^\alpha}{z^\alpha}dv + \int_{\gamma z}^{\frac{1}{2}}\frac{1-\cos v}{v^{1+\alpha}}\frac{v^\alpha}{z^\alpha}\max\{1,\frac{E[|Y|^\beta]}{(\frac{v}{z})^\beta}\}dv \\
&\quad+\int_{\frac{1}{2}}^{\infty}\frac{1-\cos v}{v^{1+\alpha}}\frac{v^\alpha}{z^\alpha}\max\{1,\frac{E[|Y|^\beta]}{(\frac{v}{z})^\beta}\}dv) \\
&\leq 2\gamma^\alpha\int_0^{\gamma z}\frac{1-\cos v}{v^{1+\alpha}}dv + 2\int_{\gamma z}^{\frac{1}{2}}\frac{1-\cos v}{v^\infty}\frac{v^\alpha}{z^\alpha}\frac{E[|Y|^\beta]}{(\frac{v}{z})^\beta}dv \\
&\leq \gamma^\alpha\int_0^{\gamma z}\frac{v^2}{v^{1+\alpha}}dv + E[|Y|^\beta](\int_{\gamma z}^{\frac{1}{2}}\frac{v^2}{v^{1+\alpha}}\left(\frac{v}{z}\right)^{\alpha-\beta}dv + 4\int_{\frac{1}{2}}^{\infty}\frac{1}{v^{1+\alpha}}\left(\frac{v}{z}\right)^{\alpha-\beta}dv) \\
&= \gamma^\alpha\int_0^{\gamma z}v^{1-\alpha}dv + E[|Y|^\beta]z^{\beta-\alpha}(\int_{\gamma z}^{\frac{1}{2}}v^{1-\beta}dv + 4\int_{\frac{1}{2}}^{\infty}v^{-1-\beta}dv)
\end{aligned}$$

$$
\begin{aligned}
&= \gamma^\alpha \frac{1}{2-\alpha}(\gamma z)^{2-\alpha} + E[|Y|^\beta] z^{\beta-\alpha}\left(\frac{1}{2-\beta}\left(\frac{1}{2^{2-\beta}} - (\gamma z)^{2-\beta}\right) + \frac{4}{\beta}\left(\frac{1}{2^\beta}\right)\right) \\
&= z^{\beta-\alpha}\left(\frac{\gamma^2 z^{2-\beta}}{2-\alpha} + E[|Y|^\beta]\left(\frac{\frac{1}{2^{2-\beta}} - (\gamma z)^{2-\beta}}{2-\beta} + \frac{4}{\beta 2^\beta}\right)\right)
\end{aligned}
$$

Since $\alpha < \beta < 2$, all the z terms above tend to zero as z gets close to 0. This shows that $\Re(\Psi(s))$ is $O(s^\alpha)$ near the origin. Now for the imaginary part, we follow a similar argument. In the middle of it, we will need to consider the distribution for $B = -Y$ which we will denote by G.

$$
\begin{aligned}
\left|\frac{\mathrm{Im}(\Psi(z))}{z^\alpha}\right| &= \left|\frac{\mathrm{Im}(f(z))}{z^\alpha}\right| \\
&= \left|z^{-\alpha}\left(\int_{-\infty}^{\infty} \sin zy dF(y) - 0\right)\right| \\
&= \left|z^{-\alpha}\left(\int_{-\infty}^{\infty} \sin zy dF(y) - z\int_{-\infty}^{\infty} y dF(y)\right|\right. \\
&= \left|z^{-\alpha}\int_{-\infty}^{\infty} (zy - \sin zy) dF(y)\right| \\
&= \left|z^{-\alpha}\left(\int_0^\infty (zy - \sin zy) dF(y) - \int_\infty^0 (zb - \sin zb) dG(b)\right)\right| \\
&= \left|z^{-\alpha}\left(\int_0^\infty \int_0^{zy} (1-\cos u) du dF(y) + \int_0^\infty \int_0^{zb} (1-\cos u) du dG(b)\right)\right|
\end{aligned}
$$

Once again, the region of integration ($\{(u,y)\}$ such that $y \geq 0, u \geq 0, u \leq zy$) can also be expressed as $\{(u,y)\}$ such that $y \geq 0, u \geq 0, y \geq \frac{u}{z}$. So:

$$
\begin{aligned}
\left|\frac{\mathrm{Im}(\Psi(z))}{z^\alpha}\right| &= \left|z^{-\alpha}\left(\int_0^\infty \int_{\frac{u}{z}}^\infty 1 - \cos u dF(y) du + \int_0^\infty \int_{\frac{u}{z}}^\infty 1 - \cos u dG(b) du\right)\right| \\
&= \left|z^{-\alpha}\int_0^\infty (1-\cos u)\left(\int_{\frac{u}{z}}^\infty dF(y) + \int_{\frac{u}{z}}^\infty dG(b)\right) du\right| \\
&= \left|z^{-\alpha}\int_0^\infty (1-\cos u) P(|Y| \geq \frac{u}{z}) du\right| \\
&= \left|\int_0^\infty \frac{1-\cos u}{u^\alpha}\left(\frac{u}{z}\right)^\alpha P(|Y| \geq \frac{u}{z}) du\right| \\
&\leq \left|\int_0^\infty \frac{1-\cos u}{u^\alpha}\left(\frac{u}{z}\right)^\alpha \frac{E[|Y|^\beta]}{(\frac{u}{z})^\beta} du\right| \\
&= E[|Y|^\beta]|z^{\beta-\alpha}\int_0^\infty \frac{1-\cos u}{u^\beta} du| \\
&= E[|Y|^\beta]|z^{\beta-\alpha}\left(\int_0^{\frac{1}{2}} \frac{1-\cos u}{u^\beta} du + \int_{\frac{1}{2}}^\infty \frac{1-\cos u}{u^\beta} du\right)| \\
&\leq E[|Y|^\beta]\left(|z^{\beta-\alpha}\int_0^{\frac{1}{2}} \frac{1}{2}u^{2-\beta} du| + |z^{\beta-\alpha}\int_{\frac{1}{2}}^\infty 2u^{-\beta} du|\right) \\
&= E[|Y|^\beta]|z^{\beta-\alpha}|\left(\frac{1}{2(3-\beta)2^{3-\beta}} + \frac{2}{\beta-1}2^{\beta-1}\right)
\end{aligned}
$$

Since $\alpha < \beta$, the z term above tends to zero as z gets close to 0. This shows that $\mathrm{Im}(\Psi(s))$ is $O(s^\alpha)$ near the origin and since the real part is also, that $\Psi(s)$ is $O(s^\alpha)$ near the origin.

Let $S_N = \sum_{j=1}^N Y_j$. Let $G(s)$ be its distribution and let $\Phi(z)$ be its characteristic function. We now use an argument exactly like the one used to prove the "truncation inequality" in [44].

$$
\begin{aligned}
P(|S_N| > \Delta) &= \int_{|s|\geq\Delta} dG(s) \\
&= \frac{1}{1-\sin 1}\int_{|s|\geq\Delta}(1-\sin 1)dG(s) \\
&\leq \frac{1}{1-\sin 1}\int_{|s|\geq\Delta}(1-\frac{\sin\frac{s}{\Delta}}{\frac{s}{\Delta}})dG(s) \\
&\leq \frac{1}{1-\sin 1}(\int_{|s|\geq\Delta}(1-\frac{\sin\frac{s}{\Delta}}{\frac{s}{\Delta}})dG(s)+\int_{|s|<\Delta}(1-\frac{\sin\frac{s}{\Delta}}{\frac{s}{\Delta}})dG(s)) \\
&= \frac{\Delta}{1-\sin 1}\int_{-\infty}^{\infty}\frac{1}{\Delta}-\frac{\sin\frac{s}{\Delta}}{s}dG(s) \\
&= \frac{\Delta}{1-\sin 1}\int_{-\infty}^{\infty}\int_0^{\frac{1}{\Delta}}(1-\cos zx)dzdG(s) \\
&= \frac{\Delta}{1-\sin 1}\int_0^{\frac{1}{\Delta}}\int_{-\infty}^{\infty}(1-\cos zx)dG(s)dz \\
&= \frac{\Delta}{1-\sin 1}\int_0^{\frac{1}{\Delta}}\Re(\Psi_N(z))dz
\end{aligned}
$$

Where $\Psi_N(z)$ is defined by the following:

$$
\begin{aligned}
\Phi(z) &= \int e^{iz\sum j=1^N y_j}dF(y_1)dF(y_2)\cdots dF(y_N) \\
&= \int\prod_{j=1}^N e^{izy_j}dF(y_1)dF(y_2)\cdots dF(y_N) \\
&= \prod_{j=1}^N\int_{-\infty}^{\infty}e^{izy_j}dF(y_j) \\
&= (f(z))^N \\
&= (1-\Psi(z))^N \\
&= 1-\Psi_N(z)
\end{aligned}
$$

We are interested in large deviations Δ and hence in small $z \in (0, \frac{1}{\Delta})$. In a neighborhood of the origin, since $\Psi(z)$ is $O(s^\alpha)$, we know that there is a K such that as long as $|z| < \delta$ we have $|\Re(\Psi(z))| \leq K|z|^\alpha$ and $|\mathrm{Im}(\Psi(z))| \leq K|z|^\alpha$ as well. It is clear from the simple binomial expansion that as long as $|z| < \delta'$, $\Re(\Psi_N(z)) \leq 2NK|z|^\alpha$ as well. Focusing our attention on $\Delta > \frac{1}{\delta'}$, we have:

$$
\begin{aligned}
P(|S_N| \geq \Delta) &\leq \frac{\Delta}{1-\sin 1}\int_0^{\frac{1}{\Delta}}\Re(\Psi_N(z))dz \\
&\leq \frac{\Delta}{1-\sin 1}\int_0^{\frac{1}{\Delta}}2NKz^\alpha dz
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{2NK\Delta}{(1-\sin 1)(\alpha+1)}\left(\frac{1}{\Delta}\right)^{\alpha+1} \\
&= \frac{2NK\Delta^{-\alpha}}{(1-\sin 1)(\alpha+1)}
\end{aligned}
$$

Finally, we know that:

$$
\begin{aligned}
P(|\frac{1}{N}\sum_{i=1}^{N} X_i - \bar{X}| \geq \epsilon) &= P(|\frac{S_N}{N}| \geq \epsilon) \\
&= P(|S_N| \geq N\epsilon) \\
&\leq \frac{2NK(N\epsilon)^{-\alpha}}{(1-\sin 1)(\alpha+1)} \\
&= \frac{2K}{(1-\sin 1)(\alpha+1)\epsilon^{\alpha}} N^{1-\alpha}
\end{aligned}
$$

Which proves the theorem. $\square$.

Notice that all we use in the proof of Theorem B.2.1 is that $P(X_i \geq \Delta) \leq K\Delta^{-\beta}$ where $\beta \in (1,2)$. It is obvious that a mean $\bar{X}$ exists since:

$$
\begin{aligned}
E[X] &= \int_0^{\infty} P(X > x)dx \\
&\leq \int_0^{K^{\frac{1}{\beta}}} P(X > x)dx + \int_{K^{\frac{1}{\beta}}}^{\infty} Kx^{-\beta}dx \\
&\leq K^{\frac{1}{\beta}} + K\int_{K^{\frac{1}{\beta}}}^{\infty} x^{-\beta}dx \\
&\leq K^{\frac{1}{\beta}} + \frac{K}{\beta-1}\left(K^{\frac{1}{\beta}}\right)^{1-\beta} \\
&= \frac{\beta K^{\frac{1}{\beta}}}{\beta-1} < \infty
\end{aligned}
$$

Thus we immediately have the following corollary:

Corollary B.2.1 *Given i.i.d. non-negative random variables $\{X_i\}$ for which there exists a constant K so that $P(X_i \geq \Delta) \leq K\Delta^{-\beta}$ for $\beta \in (1,2)$, the empirical average $\frac{1}{N}\sum_{i=1}^{N} X_i$ converges to a finite $\bar{X}$ as follows: $\forall \alpha \in (1,\beta)$ there exists a distribution dependent constant K' such that:*

$$
P(|\frac{1}{N}\sum_{i=1}^{N} X_i - \bar{X}| \geq \epsilon) \leq \frac{K'}{\epsilon^{\alpha}} N^{1-\alpha}
$$

Appendix C

“Simulating” Random Variables

In this appendix, we will collect some technical results that are used in Chapters 4 and 6. These have to do with a constructing random variable X out of other independent random variables so that X has a given distribution F.

C.1 Definitions and Examples

Because linear systems are of particular interest to us, we are interested in being able to make random variables by taking sums of other random variables.

Definition C.1.1 *A distribution F is* additively constructible out of distribution G *if there exists a distribution G' such that if Y is a random variable with distribution G, and Y' an independent one with distribution G', then $X = Y + Y'$ has distribution F (or differs from F only on a zero-measure set).*

Our most basic distributions for random variables are the Dirac δ distribution and the uniform distribution.

Example C.1.1 *Consider real-valued random variables. Suppose that $G(x) = \frac{1}{2}\delta(x - \frac{\theta}{4}) + \frac{1}{2}\delta(x + \frac{\theta}{4})$ with δ being the standard Dirac distribution. Then, a distribution F with a uniform density function f that has support on $[0, \theta)$ is additively constructible out of distribution G. The appropriate G' is defined by its density g' as follows:*

$$g'(x) = \begin{cases} \frac{2}{\theta} & \text{if } x \in [\frac{\theta}{4}, \frac{3\theta}{4}] \\ 0 & \text{otherwise} \end{cases}$$

Example C.1.2 *Consider real-valued random variables. Suppose that G is a uniform random variable on $[0, \theta)$. Then any distribution F with a piecewise constant density function f that is constant on domains on the form $[k\theta, (k+1)\theta)$ is additively constructible out of distribution G. The appropriate G' is:*

$$G'(x) = \frac{1}{\theta} \sum_{t=-\infty}^{\infty} f(t + \frac{\theta}{2})\delta(x - t\theta)$$

But often, the type of exact matches illustrated by examples C.1.1 or C.1.2 are not possible.

Definition C.1.2 *A distribution F is* ϵ-approximately additively constructible out of distribution G *if there exists distributions G' and H such that if (Y, Y', Z, B_ϵ) are independent random variables with Y having distribution G, Y' with distribution G', Z with distribution H, and B_ϵ Bernoulli distribution with $P(1) = \epsilon$, then $X = (1 - B_\epsilon)(Y + Y') + B_\epsilon Z$ has distribution F (or differs from F only on a zero-measure set).*

The idea here is that F can be approximately constructed out of G and G', with the residual coming from the occasional replacement by Z. Procedurally, we would generate a sample X as follows:

- Generate an independent sample $b = B_\epsilon$. If $b = 1$ goto step 4
- Generate independent samples $y = Y$ and $y' = Y'$.
- Assign $x = y + y'$ and return.
- Generate independent sample $z = Z$.
- Assign $x = z$ and return.

Clearly, by using $\epsilon = 0$, we are back to the case of additively constructed since we will never reach step 4 in the above procedure.

Example C.1.3 *Consider real-valued random variables. Suppose that G is a uniform random variable on $[0, \theta)$. Then a distribution F with density $f(x) = \frac{1}{\theta} - \frac{1}{2\theta^2}x$ on the interval $[0, 2\theta]$ and $f(x) = 0$ outside of that interval is $\frac{1}{2}$-approximately additively constructible out of the uniform distribution on $[0, \theta]$ by using $G'(x) = \delta(x)$ and $H(x)$ defined by its density as:*

$$h(x) = \begin{cases} \frac{1}{\theta} - \frac{1}{\theta^2}x & \text{if } x \in [0, \theta) \\ \frac{1}{\theta} - \frac{1}{\theta^2}(x - \theta) & \text{if } x \in [\theta, 2\theta] \\ 0 & \text{otherwise} \end{cases}$$

Often, we will be interested in letting ϵ get arbitrarily small by using a parametric family of distributions for G and varying their parameter θ.

Definition C.1.3 *A distribution F is* arbitrarily additively constructible out of a θ-parametric family of distributions G *if there exists a sequence of $\{(\epsilon_i, \theta_i)\}$ with $\lim_{i\to\infty} \epsilon_i = 0$ such that F is ϵ_1-approximately additively constructible out of distribution G^{θ_i}.*

The triangular random variable from example C.1.3 fits this definition relative to the family of uniform densities indexed by the width of support.

Example C.1.4 *Consider real-valued random variables. Suppose that G^θ is a uniform random variable on $[0, \theta)$. Then the distribution F with density $f(x) = \frac{1}{a} - \frac{1}{2a^2}x$ on the interval $[0, 2a]$ and $f(x) = 0$ outside of that interval is arbitrarily additively constructible out of uniform distributions on $[0, \theta]$.*

We use a sequence with $(\epsilon_i, \theta_i) = (\frac{1}{2^{i+1}}, \frac{a}{2^i})$ and at the i-th approximation, we use:

$$G'(x) = \frac{2}{2^{i+1} - 1} \sum_{j=1}^{2^{i+1}-1} (1 - \frac{j}{2^{i+1}})\delta(x - \frac{a(j-1)}{2^i})$$

and $H(x)$ defined by its density as:

$$h(x) = \begin{cases} \frac{1}{a} - \frac{1}{2^i a^2}(x - (j-1)\frac{a}{2^i}) & \textit{if } x \in [(j-1)\frac{a}{2^i}, j\frac{a}{2^i}) \textit{ and } 1 \le j \le 2^{i+1} - 1 \\ 0 & \textit{otherwise} \end{cases}$$

C.2 Results

With the definitions out of the way, we can state some elementary results that we need in the rest of the thesis. The first property allows us to build complicated distributions out of simpler ones.

Lemma C.2.1 *If distribution E is additively constructible out of distribution F, and F is additively constructible out of distribution G, then E is additively constructible out of distribution G as well.*

Similarly, if distribution E is arbitrarily additively constructible out of the θ-parametric family of distribution F, and each F^θ is additively constructible out of distribution G^θ, then E is arbitrarily additively constructible out of the θ-parametric family of distributions G as well.

Proof: X with distribution E can be written as $X = Y + Y'$ with the Y having distribution F. But, Y can be written as $Y = Z + Z'$ with Z having distribution G. Then $X = Z + (Z' + Y')$ and hence E is additively constructible out of distribution G.

The statement for arbitrarily constructible follows immediately. □

Example C.1.1 tells us one way of generating a uniform random variable. We will also need a way of getting i.i.d. uniform random variables by means of "dithering" other random variables with possibly complicated dependencies and distributions.

Lemma C.2.2 *Given any real valued random process $\{X_t\}$, and an independent sequence of i.i.d. uniform random variables $\{\Psi_t\}$ of width θ. Consider processes $\{(M_t, \Phi_t)\}$ that satisfy:*

$$M_t\theta + \Phi_t = X_t + \Psi_t$$

with M_t taking values on the integers and ϕ_t taking values in $[0, \theta)$. Then:

1. *Knowing $\Psi_t = \psi$, $M_t = m$, and $\Phi_t = \phi$ allows us to uniquely determine the value of X_t.*

2. *Knowing $X_t = x$ and $\Psi_t = \psi$ allows us to uniquely determine the values of (M_t, Φ_t).*

3. *$\{\Phi_t\}$ is an i.i.d. sequence of uniform random variables of width θ even after conditioning on all random variables in the system except any that are dependent on Ψ_t itself.*

Proof: The first claim is obvious since $P(X_t = x \,|\, \Psi_t = \psi, M_t = m, \Phi_t = \phi) = \delta(x - m\theta - \phi - \psi)$ by definition. For the second, consider $m\theta + \phi = x + \psi = m'\theta + \phi'$. Dividing both sides by θ gives us $m + \frac{\phi}{\theta} = m' + \frac{\phi'}{\theta}$. But $\frac{\phi}{\theta}$ and $\frac{\phi'}{\theta}$ are both in $[0, 1)$. Since the integral and proper fractional parts of a real number are uniquely determined, this means that

$$m = m' = \left\lfloor \frac{x}{\theta} + \frac{\psi}{\theta} \right\rfloor = \left\lfloor \frac{x}{\theta} \right\rfloor + \left\lfloor (\frac{x}{\theta} - \left\lfloor \frac{x}{\theta} \right\rfloor) + \frac{\psi}{\theta} \right\rfloor$$

and thus $\phi = \phi'$ also holds.

$$\begin{aligned}
& P(\Phi_t \le \phi \Big| X_1^\infty = x_1^\infty, \Psi_1^{t-1} = \psi_1^{t-1}, \Psi_{t+1}^\infty = \psi_{t+1}^\infty, \\
& M_1^{t-1} = m_1^{t-1}, M_{t+1}^\infty = m_{t+1}^\infty, \Phi_1^{t-1} = \phi_1^{t-1}, \Phi_{t+1}^\infty = \phi_{t+1}^\infty) \\
= \;& P(\frac{\Phi_t}{\theta} \le \frac{\phi}{\theta} \Big| X_1^\infty = x_1^\infty, \Psi_1^{t-1} = \psi_1^{t-1}, \Psi_{t+1}^\infty = \psi_{t+1}^\infty) \\
= \;& P(\frac{x_t + \Psi_t}{\theta} - M_t \le \frac{\phi}{\theta} \Big| X_1^\infty = x_1^\infty, \Psi_1^{t-1} = \psi_1^{t-1}, \Psi_{t+1}^\infty = \psi_{t+1}^\infty) \\
= \;& P(\frac{x_t}{\theta} + \frac{\Psi_t}{\theta} - \left\lfloor \frac{x_t}{\theta} \right\rfloor - \left\lfloor (\frac{x_t}{\theta} - \left\lfloor \frac{x_t}{\theta} \right\rfloor) + \frac{\Psi_t}{\theta} \right\rfloor \le \frac{\phi}{\theta} \Big| X_1^\infty = x_1^\infty, \Psi_1^{t-1} = \psi_1^{t-1}, \Psi_{t+1}^\infty = \psi_{t+1}^\infty) \\
= \;& P(\frac{\Psi_t}{\theta} + (\frac{x_t}{\theta} - \left\lfloor \frac{x_t}{\theta} \right\rfloor) - \left\lfloor (\frac{x_t}{\theta} - \left\lfloor \frac{x_t}{\theta} \right\rfloor) + \frac{\Psi_t}{\theta} \right\rfloor \le \frac{\phi}{\theta} \Big| X_1^\infty = x_1^\infty, \Psi_1^{t-1} = \psi_1^{t-1}, \Psi_{t+1}^\infty = \psi_{t+1}^\infty)
\end{aligned}$$

To simplify the notation, let us substitute $\Gamma = \frac{\Phi}{\theta}$ and $\Pi = (\frac{X_t}{\theta} - \left\lfloor \frac{X_t}{\theta} \right\rfloor)$ (with the lower case γ and π being defined analogously for the non-random variables). By construction we know that $0 \le \Gamma < 1$ and $0 \le \Pi < 1$.

$$\begin{aligned}
& P(\Phi_t \le \phi \Big| X_1^\infty = x_1^\infty, \Psi_1^{t-1} = \psi_1^{t-1}, \Psi_{t+1}^\infty = \psi_{t+1}^\infty, \\
& M_1^{t-1} = m_1^{t-1}, M_{t+1}^\infty = m_{t+1}^\infty, \Phi_1^{t-1} = \phi_1^{t-1}, \Phi_{t+1}^\infty = \phi_{t+1}^\infty) \\
= \;& P(\frac{\Psi_t}{\theta} + \pi - \left\lfloor \pi + \frac{\Psi_t}{\theta} \right\rfloor \le \gamma \Big| \Pi_1^\infty = \pi_1^\infty, \Psi_1^{t-1} = \psi_1^{t-1}, \Psi_{t+1}^\infty = \psi_{t+1}^\infty) \\
= \;& P(\frac{\Psi_t}{\theta} + \pi - 1 \le \gamma \Big| \pi + \frac{\Psi_t}{\theta} \ge 1, \Pi_1^\infty = \pi_1^\infty, \Psi_1^{t-1} = \psi_1^{t-1}, \Psi_{t+1}^\infty = \psi_{t+1}^\infty) \\
& P(\pi + \frac{\Psi_t}{\theta} \ge 1 \Big| \Pi_1^\infty = \pi_1^\infty, \Psi_1^{t-1} = \psi_1^{t-1}, \Psi_{t+1}^\infty = \psi_{t+1}^\infty\Big) \\
& + P(\frac{\Psi_t}{\theta} + \pi \le \gamma \Big| \pi + \frac{\Psi_t}{\theta} < 1, \Pi_1^\infty = \pi_1^\infty, \Psi_1^{t-1} = \psi_1^{t-1}, \Psi_{t+1}^\infty = \psi_{t+1}^\infty) \\
& P(\pi + \frac{\Psi_t}{\theta} < 1 \Big| \Pi_1^\infty = \pi_1^\infty, \Psi_1^{t-1} = \psi_1^{t-1}, \Psi_{t+1}^\infty = \psi_{t+1}^\infty) \\
= \;& P(\frac{\Psi_t}{\theta} \ge 1 - \pi) P(\frac{\Psi_t}{\theta} \le \gamma - \pi + 1 \Big| \frac{\Psi_t}{\theta} \ge 1 - \pi) \\
& + P(\frac{\Psi_t}{\theta} < 1 - \pi) P(\frac{\Psi_t}{\theta} \le \gamma - \pi \Big| \frac{\Psi_t}{\theta} < 1 - \pi) \\
= \;& \pi \int_{1-\pi}^{\min(1-\pi+\gamma,1)} \frac{1}{\pi} d\psi + (1-\pi) \int_0^{\max(\gamma-\pi,0)} \frac{1}{1-\pi} d\psi \\
= \;& (\min(1-\pi+\gamma, 1) - (1-\pi)) + (\max(\gamma - \pi, 0) - 0) \\
= \;& \pi + \min(\gamma - \pi, 0) + \max(\gamma - \pi, 0) \\
= \;& \pi + \gamma - \pi \\
= \;& \frac{\phi}{\theta}
\end{aligned}$$

This shows that Φ, even after conditioning on specific values for all other random variables (except Ψ_t and M_t) in the system, is a uniform random variable on $[0, \theta)$. Since it is uniform for all specific values for these values, it remains uniform when we integrate over their measures and remove the conditioning. Thus, the sequence $\{\Phi_t\}$ is an i.i.d. sequence of uniform random variables of width θ. □

The third property states the immediate fact that the operation of linear systems distribute over addition.

Lemma C.2.3 *Consider an i.i.d. process $\{W_t\}$ with distribution F for each W_t that is fed into a discrete-time linear system L to give output process $\{X_t\}$. If we realize each W_t as an ϵ-approximately additively constructible variable out of (Y_t, Y_t', Z_t, B_t), then we can express X_t as a sum of two terms: one involving the linear system's response to $\{(1 - B_t)Y_t\}$, and the other involving the linear system's response to $\{(1 - B_t)Y_t' + B_t Z_t\}$.*

Proof: Obvious by the definition of linear system. □

We will also want to know some information theoretic consequences of our approximation strategy which consists of viewing one random variable as the convex combination of two random variables.

Lemma C.2.4 *Suppose X is a random variable with distribution F which can be represented $X = (1 - B_\epsilon)Y + B_\epsilon Z$ where B_ϵ is a Bernoulli variable with head probability ϵ and (Y, Z) and B_ϵ are independent. Let $p(\hat{X}|X)$ be a transition measure and the pair $(X, \hat{X})$ have measure defined by $F(X)p(\hat{X}|X)$. Define the pairs $(Y, \check{X})$ and $(Z, \tilde{X})$ by the analogous combinations of the transition measure p with the measures for Y and Z respectively. If $\rho(X, \hat{X}) \geq 0$ is a distort measure, then:*

$$E[\rho(Y, \check{X})] \leq \frac{1}{1-\epsilon} E[\rho(X, \hat{X})]$$

and also

$$I(Y; \check{X}) \leq \frac{1}{1-\epsilon} I(X, \hat{X})$$

where I is the information theoretic mutual information function.

Proof: The first part is a simple consequence of the nonnegativity of ρ:

$$\begin{aligned} E[\rho(X, \hat{X})] &= (1-\epsilon)E[\rho(Y, \check{X})] + \epsilon E[\rho(Z, \tilde{X})] \\ &\geq (1-\epsilon)E[\rho(Y, \check{X})] \end{aligned}$$

which immediately yields the desired result by dividing both sides by $(1 - \epsilon)$.

As far as the mutual information goes, we rely on the concavity of mutual information (Theorem 2.7.4 of [11]) to give:

$$\begin{aligned} I(X; \hat{X}) &\geq (1-\epsilon)I(Y; \check{X}) + \epsilon I(Z; \tilde{X}) \\ &\geq (1-\epsilon)I(Y; \check{X}) \end{aligned}$$

where the first inequality expresses concavity of $I(X, \hat{X})$ with respect to the measure for X and the second inequality comes from the fact that mutual information is always positive. From this, we can again get our desired result by dividing both sides by $(1 - \epsilon)$. □

We also need to understand which sort of distributions can be arbitrarily additively constructed from other distributions:

Theorem C.2.1 *If probability distribution F has a bounded density f which is Riemann-Integrable on any bounded set and is continuous outside of that bounded set, then F can* *be arbitrarily additively constructed from uniform random variables parametrized by their widths.*

Proof: Follows immediately from the definition of Riemann integrability since the lower Riemann sums corresponding to the integral of the probability measure will approach 1. The Riemann sums correspond to the sum of an appropriately narrow uniform random variable to an appropriately modulated train of delta functions. Since the sums approach 1, it means that ϵ can be made to go to 0. $\square$

We will sometime also need to be able to bound the required relationship between ϵ_i and θ_i as we make the ϵ_i go to zero.

Theorem C.2.2 *If probability distribution F has a bounded density f with bounded support which is Lipshitz everywhere (ie. $\exists L$ such that $\forall x, x'$ we have $|f(x) - f(x')| \leq L|x - x'|$), then F can be arbitrarily additively constructed from uniform random variables parametrized by their widths θ. Furthermore, there exists a constant K so that we can choose (ϵ_i, θ_i) with $\epsilon_i \leq KL\theta_i$.*

Proof: Let the support of f be within $[x_0, x_0 + \Omega]$. Then, if we choose $\theta = \frac{\Omega}{M}$, we can use the following to approximate f.

$$g(x) = \begin{cases} \inf_{x' \in [x_0 + \frac{\Omega}{M}(j-1), x_0 + \frac{\Omega}{M}j]} f(x') & \text{if } x \in [x_0 + \frac{\Omega}{M}(j-1), x_0 + \frac{\Omega}{M}j] \text{ and } 1 \leq j \leq M \\ 0 & \text{otherwise} \end{cases} \tag{C.1}$$

It should be clear how (C.1) can be realized as the density representing the sum of a uniform random variable with support $\theta = \frac{\Omega}{M}$ and a suitable sum of modulated δ functions. The only question is how much probability is left over.

$$\begin{aligned} \epsilon &= \int_{x_0}^{x_0+\Omega} f(x) - g(x)dx \\ &= \sum_{j=1}^{M} \int_{x_0+\frac{\Omega}{M}(j-1)}^{x_0+\frac{\Omega}{M}j} \left(f(x) - \inf_{x' \in [x_0+\frac{\Omega}{M}(j-1), x_0+\frac{\Omega}{M}j]} f(x') \right) dx \\ &\leq \sum_{j=1}^{M} \frac{L}{2} (\frac{\Omega}{M})^2 \\ &= \frac{\Omega}{2} L \frac{\Omega}{M} \\ &= \frac{\Omega}{2} L\theta \end{aligned}$$

The inequality comes from the fact that the function is Lipshitz and thus the area cannot exceed that of a triangle with slope L. $\square$

These results for uniform random variables naturally generalize to the pairs of δ functions separated by θ from Example C.1.1 by the application of Lemma C.2.1.

Bibliography

[1] R. Bansal and T. Basar, "Solutions to a class of linear-quadratic-Gaussian (LQG) stochastic team problems with nonclassical information." System and Control Letters 9:125–130, 1987.

[2] T. Basar, "A Trace Minimization Problem with Applications in Joint Estimation and Control under Nonclassical Information." Journal of Optimization Theory and Applications, 31(3):343–359, 1980.

[3] T. Basar and R. Bansal, "Optimum design of measurement channels and control policies for linear-quadratic stochastic systems." European Journal of Operational Research 73:226–236, 1994.

[4] L. A. Bassalygo and V. V. Prelov, "Zero-error capacity of continuous channels without storage with a small input signal." Problemy Peredachi Informatsii, Volume 17, No. 1, pp 5-16, 1981.

[5] T. Berger, "Information Rates of Wiener Processes." IEEE Transactions on Information Theory, 16(2):134–139, 1970.

[6] T. Berger, *Rate Distortion Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1971.

[7] T. Berger, "Lossy Source Coding." IEEE Transactions on Information Theory, 44(6):2693–2723, 1998.

[8] K. G. Binmore and H. H. Stratton, "A Note on Characteristic Functions." Annals of Mathematical Statistics., 40(1)303–307, 1969.

[9] Vivek Borkar, Sanjoy Mitter, Anant Sahai, and Sekhar Tatikonda, "Sequential Source Coding: An Optimization Viewpoint" Submitted to IEEE Transactions on Information Theory

[10] A. Bogomolny, "Infinity and Probability." URL: http://www.cut-the-knot.com/Probability/infinity.html, 2000.

[11] T. Cover and J. Thomas, *Elements of Information Theory.* John Wiley, New York, 1991

[12] P. Diaconis and D. Freedman "Iterated Random Functions." SIAM Review, 41(1)45–76, 1999

[13] R. L. Dobrushin, "General formulation of Shannon's main theorem of information theory." Usp. Math. Nauk., 14:3-104, 1959. Translated in Am. Math. Soc. Trans., 33:323-438.

[14] J. Doyle and J. M. Carlson, "Power Laws, Highly Optimized Tolerance, and Generalized Source Coding." Physical Review Letters, 84(24):5656–5659, 2000.

[15] A. Ephremides and B. Hajek, "Information Theory and Communication Networks: An Unconsummated Union." IEEE Trans. on Information Theory, 44(6):2416–2434, 1998.

[16] W. H. R. Equitz and T. Cover, "Successive Refinement of Information." IEEE Transactions on Information Theory, 37(2):269–275, 1991.

[17] F. Faltin, N. Metropolis, B. Ross, and G. C. Rota, "The Real Numbers as a Wreath Product." Advances in Mathematics 16(3):279–304, 1975.

[18] G. D. Forney, Jr., *Concatenated Codes*. Cambridge, MA: M.I.T. Press, 1966.

[19] G. D. Forney, Jr., "Convolutional codes II. Maximum-likelihood decoding." Inform. and Control, vol. 25 pp. 222-266, 1974

[20] G. D. Forney, Jr., "Convolutional codes III. Sequential decoding." Inform. and Control, vol. 25 pp. 267-297, 1974

[21] E. Fujiwara, H. Chen, and M. Kitakami, "Error recovery for ziv-lempel coding by using UEP schemes." International Symposium on Information Theory, Proceedings. IEEE pg 429. 2000

[22] G. Gabor and Z. Gyorfi, *Recursive Source Coding*. New York, NY: Springer-Verlag, 1986.

[23] R. Gallager, *Information Theory and Reliable Communication*. New York, NY: John Wiley and Sons, 1971.

[24] R. Gallager, *Information Theory and Reliable Communication*. Notes from CISM, Udine, Italy: Springer-Verlag, 1972.

[25] R. Gallager, *Discrete Stochastic Processes*. Boston, MA: Kluwer Academic Publishers, 1996.

[26] M. Gardner, *Aha! Gotcha: Paradoxes to Puzzle and Delight*. San Francisco, CA: W. H. Freeman, 1982.

[27] A. K. Gorbunov and M. S. Pinsker, "Nonanticipatory and Prognostic Epsilon Entropies and Message Generation Rates," Problemy Peredachi Informatsii, Volume 9, No. 3, pp 12–21, 1973.

[28] R. Gray, D. L. Neuhoff, and J. K. Omura, "Process Definitions of Distortion-Rate Functions and Source Coding Theorems." IEEE Transactions on Information Theory, 21(5):524–532, 1975.

[29] R. Gray, "Information Rates of Autoregressive Processes." IEEE Transactions on Information Theory, 16(4):412–421, 1970.

[30] T. S. Han, "An information-spectrum approach to source coding theorems with a fidelity criterion." IEEE Trans. on Information Theory, Volume 43, pp 1145 - 1164, July 1997.

[31] T. Hashimoto and S. Arimoto, "On the Rate-Distortion Function for the Nonstationary Autoregressive Process." IEEE Transactions on Information Theory, 26(4):478–480, 1980.

[32] P. Haskell, D. G. Messerschmitt, and L. Yun, "Architectural Principles for Multimedia Networks," in *Wireless Communications: Signal Processing Perspectives*, H. V. Poor and G. W. Wornell, editors, Upper Saddle River, NJ: Prentice-Hall, 1998.

[33] B. Hochwald and K. Zeger, "Tradeoff Between Source and Channel Coding." IEEE Trans. on Information Theory, 43(5)1412–1424, 1997.

[34] M. Horstein, "Sequential Transmission Using Noiseless Feedback." IEEE Transactions on Information Theory, Vol 12, pp.448–455, October 1966.

[35] A. K. Jain, *Fundamentals of Digital Image Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[36] R. E. Kalman. *New methods and results in linear prediction and filtering theory.* RIAS Report 61-1, RIAS, 1960.

[37] J. C. Kieffer, "A Survey of the Theory of Source Coding with a Fidelity Criterion." IEEE Transactions on Information Theory, 39(5):1473–1490, 1993.

[38] K. Korner and A. Orlitsky, "Zero-Error Information Theory." IEEE Transactions on Information Theory, 44(6):2207–2229, 1998.

[39] G. Kramer, "Directed Information for Channels with Feedback." Ph.D. thesis, Swiss Federal Institute of Technology, Zurich, 1998.

[40] S. I. Krich, "Coding for a Time Varying Fidelity Criterion." PhD Dissertation, Cornell University, 1972.

[41] S. I. Krich and T. Berger, "Coding for a Delay-Dependent Fidelity Criterion." IEEE Transactions on Information Theory, 20(1):77–85, 1974.

[42] S. I. Krich, "Asymptotic Properties of Delay-Time-Weighted Probability of Error." IEEE Transactions on Information Theory, 20(3):278–279, 1974.

[43] K.H. Lee and D.P. Petersen, "Optimal Linear Coding for Vector Channels." IEEE Transactions on Communications, Volume 24, No. 12, pp 1283-1290, December 1976

[44] M. Loeve, *Probability Theory.* Princeton, NJ: D. Van Nostrand Company, 1962

[45] G. Nair and R. Evans, "State Estimation with a Finite Data Rate." Forthcoming paper, 1998

[46] J. Nilsson, "Real-Time Control Systems with Delays." PhD Dissertation, Lund, Sweden, 1998.

[47] B. Masnick and J. Wolf, "On Linear Unequal Error Protection Codes." IEEE Transactions on Information Theory, 3(4):600–607, 1967

[48] David L. Neuhoff and R. Kent Gilbert, "Causal Source Codes," IEEE Transactions on Information Theory, 28(5):701–713, 1982.

[49] I. A. Ovseevich and M. S. Pinsker, "Linear Transmission of Nonstationary Gaussian Messages in a Gaussian Channel with Feedback." Problemy Peredachi Informatsii, Volume 10, No. 1, pp 3–8, 1974.

[50] M. S. Pinsker, "Bounds of the Probability and of the Number of Correctable Errors for Nonblock Codes," Problemy Peredachi Informatsii, Volume 3, No. 4, pp 58–71, 1967.

[51] J. Postel, "Transmission Control Protocol", RFC 793, USC/Information Sciences Institute, September 1981 (http://www.ietf.org/rfc/rfc0793.txt)

[52] S. Rajagopalan, "A Coding Theorem for Distributed Computation." PhD Dissertation, University of California at Berkeley, 1994

[53] B. Rimoldi, "Successive Refinement of Information: Characterization of the Achievable Rates." IEEE Transactions on Information Theory, 40(1):253–259, 1994.

[54] A. Sahai, "Information and Control." Unpublished Area Exam, Massachusetts Institute of Technology, December 1997

[55] A. Sahai, S. Tatikonda, S. Mitter, "Control of LQG Systems Under Communication Constraints." Proceedings of the 1999 American Control Conference, Vol 4 pp2778–2782, 1999.

[56] A. Sahai, "Evaluating Channels for Control: Capacity Reconsidered." Proceedings of the 2000 American Control Conference, Vol 4 pp2358–2362, 2000

[57] A. Sahai, ""Any-time" Capacity and A Separation Theorem For Tracking Unstable Processes", International Symposium on Information Theory, Proceedings. pg 500, 2000

[58] L. Schulman, "Coding for Interactive Communication." IEEE Trans. on Information Theory, Vol 42, pp. 1745-1756, November 1996.

[59] J. P. M. Schalkwijk and T. Kailath, "A coding scheme for additive noise channels with feedback – I: No bandwidth constraint." IEEE Trans. on Information Theory, Vol 12, pp 172-182, April 1966

[60] C. E. Shannon, "A Mathematical Theory of Communication." Bell System Technical Journal, vol. 27, pp. 379-423, 623-656, July and October 1948.

[61] C. E. Shannon, "The Zero Error Capacity of a Noisy Channel." IEEE Trans. on Information Theory, Vol 2, pp. S8-S19, September 1956.

[62] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," IRE Nat. Conv. Rec., part 4, pp. 142-163, 1959

[63] S. Tatikonda, "Control Under Communication Constraints." PhD Dissertation, 2000.

[64] S. Tatikonda, A. Sahai, S. Mitter, "Control of LQG Systems Under Communication Constraints." Forthcoming paper.

[65] D. N. C. Tse, "Variable-rate Lossy Compression and its Effects on Communication Networks," PhD Dissertation, Massachusetts Institute of Technology, 1995.

[66] S. Vembu, S. Verdu, and Y. Steinberg, "The Source-Channel Separation Theorem Revisited." IEEE Transactions on Information Theory, 41(1):44–54, 1995.

[67] S. Verdu and T. S. Han, "A General Formula for Channel Capacity." IEEE Transactions on Information Theory, 40(4):1147–1157, 1994.

[68] V. Vinogradov, *Refined Large Deviation Limit Theorems.* Pitman Research Notes in Mathematics Series, Essex, England: Longman Scientific and Technical, 1994.

[69] H. S. Witsenhausen, "A counterexample in stochastic optimum control." SIAM Journal of Control, 6(1):131–147, 1968.

[70] H. S. Witsenhausen, "Separation of estimation and control for discrete time systems." Proceedings of the IEEE, 59(11):1557–1566, 1971.

[71] J. Wozencraft, B. Reiffen, *Sequential Decoding.* Cambridge, MA: Technology Press, 1961

[72] R. Zamir and Meir Feder, "On Universal Quantization by Randomized Uniform/Lattice Quantizers." IEEE Trans. on Information Theory. 38(2):428–436, 1992.

[73] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: a new resource ReSerVation Protocol," IEEE Network, 7(5):8–18, 1993.