Source coding and channel requirements for unstable processes

Anant Sahai, Sanjoy Mitter

sahai@eecs.berkeley.edu, mitter@mit.edu

Abstract

Our understanding of information in systems has been based on the foundation of memoryless processes. Extensions to stable Markov and auto-regressive processes are classical. Berger proved a source coding theorem for the marginally unstable Wiener process, but the infinite-horizon exponentially unstable case had been open since Gray's 1970 paper. There were also no theorems showing what is needed to transport such processes across noisy channels.

In this work, we give a fixed rate source coding theorem for the infinite-horizon problem of coding an exponentially unstable Markov process. The encoding naturally results in two distinct bitstreams that have qualitatively different QoS requirements for subsequent transport over a noisy medium. The first stream captures the information that is accumulating within the nonstationary process and requires sufficient anytime reliability on the part of any channel used to transport the process. The second part of the source-code captures the historical information that dissipates within the process and is essentially classical. A converse demonstrating the fundamentally layered nature of such sources is given by means of information-embedding ideas.

Index Terms

Nonstationary processes, rate-distortion, anytime reliability, information embedding

Department of Electrical Engineering and Computer Science at the University of California at Berkeley. A few of these results were presented at ISIT 2004 and a primitive form of others appeared at ISIT 2000 and in his doctoral dissertation.

Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology. Support for S.K. Mitter was provided by the Army Research Office under the MURI Grant: Data Fusion in Large Arrays of Microsensors DAAD19-00-1-0466 and the Department of Defense MURI Grant: Complex Adaptive Networks for Cooperative Control Subaward #03-132 and the National Science Foundation Grant CCR-0325774.

Source coding and channel requirements for unstable processes

I. Introduction

The source and channel models studied in information theory are not just interesting in their own right, but also provide insights into the architecture of reliable communication systems. Since Shannon's fundamental work, memoryless sources and channels have always been at the base of our understanding. They have provided the key insight of separating source and channel coding with the bit rate alone appearing at the interface [1], [2]. The basic story has been extended to many different sources and channels with memory for point-to-point communication [3].

However, there are still many issues for which information theoretic understanding eludes us. Networking in particular has a whole host of such issues, leading Ephremides and Hajek to entitle their survey article "Information Theory and Communication Networks: An Unconsummated Union!" [4]. 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.

Real communication networks and networked applications are quite complex. To move toward a quantitative and qualitative of understanding of such issues, tractable models that exhibit at least some of the right qualitative behavior are essential. In [5], [6], the problem of stabilization of unstable plants across a noisy feedback link was considered. There, delay and feedback considerations became intertwined and the notion of feedback anytime capacity was introduced. To stabilize an otherwise unstable plant over a noisy channel, not only is it necessary to have a channel capable of supporting a certain minimal rate, but the channel when used with noiseless feedback must also support a high enough error-exponent (called the anytime reliability) in a delay-universal fashion. This turns out to be a sufficient condition as well, thereby establishing a separation theorem for stabilization. In [7], upper bounds are given for the fixed-delay reliability functions of DMCs with and without feedback, and these bounds are shown to be tight for certain classes of channels. Moreover, the fixed-delay reliability functions with feedback are shown to be fundamentally different from (and better than) the traditional fixed-block reliability functions.

While the stabilization problem does provide certain important insights into interactive applications, the separation theorem for stabilization given in [5], [6] is coarse — it only addresses performance as a binary valued entity: stabilized or not stabilized. All that matters is the tail-behavior of the closed-loop process. To get a more refined view of the problem in terms of asymptotic performance, this paper instead consider the corresponding open-loop estimation problem. This is the seemingly classical question of lossy source coding for an *unstable* scalar Markov processes — mapping the source into bits and then seeing what is required to transport such bits using a point-to-point communication system.

A. Communication of Markov Processes

Coding theorems for stable Markov and auto-regressive processes under mean-squared-error distortion are now well established in the literature [8], [9]. We consider real-valued Markov processes, modeled as:

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

where $\{W_t\}_{t\geq 0}$ are white and X_0 is an independent initial condition uniformly distributed on $[-\frac{\Omega_0}{2}, +\frac{\Omega_0}{2}]$ where $\Omega_0 > 0$ and is small. The essence of the problem is depicted in Fig. 1: to minimize the rate of the encoding while maintaining an adequate fidelity of reconstruction. Once the source has been compressed, the resulting bitstreams can presumably be reliably communicated across a wide variety of noisy channels.

The infinite horizon source coding problem (smoothing) is to design a source code minimizing the rate R used to encode the process while keeping the reconstruction close to the original source in an average

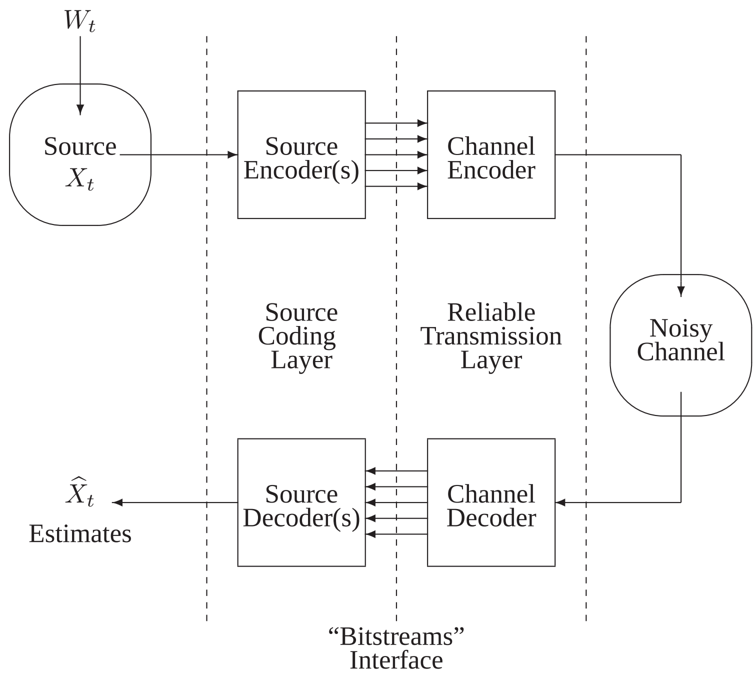

Fig. 1. The point-to-point communication problem considered here. The goal is to minimize end-to-end average distortion $\rho(X_t, \widehat{X}_t)$. Finite, but possible large, end-to-end delay will be permitted. One of the key issues explored is what must be made available at the source/channel interface.

sense[1] $\lim_{n\to\infty} \frac{1}{n}\sum_{t=1}^{n} E[|X_t - \widehat{X}_t|^\eta]$. *The key issue is that any given encoder/decoder system must have a bounded delay when used over a fixed-rate noiseless channel.* The encoder is not permitted to look into the infinite future[2] before committing to an encoding for $\widehat{X}_t$. For the stable cases $A < 1$, standard block-coding arguments work, since long blocks separated by an intervening block look relatively independent of each other. The ability to encode blocks in an independent way also tells us that Shannon's classical sense of ϵ-reliability also suffices for transporting the encoded bits across a noisy channel.

The study of unstable cases $A \geq 1$ is substantially more difficult since they are neither ergodic nor stationary and furthermore their variance grows unboundedly with time. As a result, Gray was able to prove only finite horizon results for such nonstationary processes and the general infinite-horizon unstable case had remained essentially open since Gray's 1970 paper [9]. As he put it:

> 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.

On the computational side, Hashimoto and Arimoto gave a parametric form for computing the $R(d)$ function for unstable auto-regressive Gaussian processes [10]. Toby Berger gave an explicit coding theorem for an important sub-case: the marginally unstable Wiener process ($A = 1$) by introducing an ingenious parallel stream methodology and noticing that although the Wiener process is nonstationary, it does have stationary and independent increments [11]. However, Berger's source-coding theorem said nothing about what is required from a noisy channel. In his own words:[12]

> It is worth stressing that we have proved only a source coding theorem for the Wiener process, not an

[1]For the discussion of channel requirements, the general η-moment of the difference $E[|X_t - \widehat{X}_t|^\eta]$ is the natural per-letter distortion measure. In the Gaussian case of Section V, we return to the more standard mean-squared error distortion.

[2]To allow the laws of large numbers to work, a potentially large but finite end-to-end delay is allowed between when the encoder observes X_t and when the decoder emits $\widehat{X}_t$. However, this delay is must remain bounded for all t.

> 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) that, even if a *noisy* feedback link were provided, it still would not be possible to achieve a finite [mean squared error] per letter as $t \to \infty$.

In an earlier conference work [13] and the first author's dissertation [14], we gave a variable rate coding theorem that showed that we could achieve the $R(d)$ bound in the infinite-horizon case if we were allowed the use of variable rate codes. The question of whether or not fixed rate codes could be made to work was left open.[3]

B. Asymptotic equivalences and direct reductions

Beyond the technical issue of fixed or variable rate lies a deeper question regarding the nature of "information" in such processes. [15] contains an analysis of the traditional Kalman-Bucy filter in which certain entropic expressions are identified with the accumulation and dissipation of information within a filter. No explicit source or channel coding is involved, but the idea of different kinds of information flows is raised through interpreting certain mutual information quantities. In the stabilization problem of [5], it is hard to see if any qualitatively distinct kinds of information are present since to an external observer, the closed-loop process is stable. Similarly, the variable rate code given earlier in [13], [14] also did not distinguish between kinds of information since the same high QoS requirements were imposed on all bits. However, it was unclear whether all the bits *required* the same treatment since we had given an example in which access to an additional lower reliability transport can be used to improve end-to-end performance [16], [14]. The true nature of the information within the unstable process was left open and while exponentially unstable processes certainly appeared to be accumulating information, there was no precise way to make this interpretation and quantify the amount of accumulation.

In order to understand the nature of information, this paper builds upon the "asymptotic communication problem equivalence" perspective introduced at the end of [5]. This approach associates communication problems (e.g. communicating bits reliably at rate R or communicating iid Gaussian random variables to average distortion $\leq D$) with the set of channels that are good enough to solve that problem (e.g. noisy channels with capacity $C > R$). This parallels the "asymptotic computational problem equivalence" perspective in computational complexity theory [17] except that the critical resource shifts from computational operations to noisy channel uses. The heart of the approach is the use of "reductions" that show that a system made to solve one communication problem can be used as a black box to solve another communication problem. Two problems are asymptotically equivalent if they can be reduced to each other.

The equivalence perspective is closely related to the traditional source/channel separation theorems. The main difference is that traditional separation theorems give a privileged position to one communication problem — reliable bit-transport in the Shannon sense — and use reductions in only one direction: from the source to bits. The "converse" direction is usually proved using properties of mutual information. In [18], [19], we give a direct proof of the "converse" for classical problems by showing the existence of randomized codes that embed iid data bits into iid source symbols at rate R so that the bits can be recovered with high probability from the distorted reconstructions of the source symbols as long as the average distortion on long blocks stays below the distortion-rate function $D(R)$. Similar results are obtained for the conditional distortion-rate function. This equivalence approach to separation theorems considers the privileged position of reliable bit-transport to be purely a pedagogical matter.

This paper uses the results from [18], [19] to extend the results of [5] from the control context to the estimation context. We demonstrate that the problem of communicating an unstable Markov process to

[3] For stationary processes with bounded distortion measures, a variable rate code can be easily converted into a fixed rate code by buffering and then declaring an overflow in the rare event of the variable rate code exceeding the desired fixed rate by enough to prevent an on-time delivery with the target delay. After the overflow, the code can simply restart from scratch and any distortion penalty incurred by the induced transient will be bounded and can be made to have arbitrarily low probability by choosing a buffer large enough. For unstable processes, this strategy does not work since restarting is impossible due to the information accumulating within the process.

within average distortion d is asymptotically equivalent to a pair of communication problems: classical reliable bit-transport at a rate $\approx R(d) - \log_2 A$ and anytime-reliable bit-transport at a rate $\approx \log_2 A$. This gives a precise interpretation to the nature of information flows in such processes.

C. Performance bound in the limit of large delays

To define $R(d)$ for such processes, the infinite horizon problem is viewed as the limit of a sequence of finite horizon problems:

Definition 1.1: Given the scalar Markov source given by (1), the *finite n horizon* version of the source is defined to be the random variables $X_0^{n-1} = (X_0, X_1, \ldots, X_{n-1})$.

Definition 1.2: $\eta-$distortion: The $\eta-$distortion measure is $\rho(X_i, \widehat{X}_i) = |X_i - \widehat{X}|^\eta$. It is an additive distortion measure when applied to blocks.

The standard information-theoretic rate-distortion function for the finite horizon problem using η-difference distortion is:

$$R_n^X(d) = \inf_{\{\mathcal{P}(Y_0^{n-1}|X_0^{n-1}) : \frac{1}{n}\sum_{i=0}^{n-1} E[|X_i - Y_i|^\eta] \leq d\}} \frac{1}{n} I(X_0^{n-1}; Y_0^{n-1}) \tag{2}$$

In (2), we infimize the average mutual information between X and Y over joint measures where the marginal for X_1^n is fixed and the average per-letter distortion is constrained to be below d. We can consider the block X_1^n as a single vector-valued random variable $\vec{X}$. The $R_n^X(d)$ defined by (2) is related to $R_1^{\vec{X}}(d)$ by $R_n^X(d) = \frac{1}{n} R_1^{\vec{X}}(nd)$ with the distortion measure on $\vec{X}$ given by $\rho(\vec{X}, \widehat{\vec{X}}) = \sum_{i=0}^{n-1} |X_i - \widehat{X}|^\eta$.

The infinite horizon case is then defined as a limit:

$$R_\infty^X(d) = \liminf_{n\to\infty} R_n^X(d) \tag{3}$$

The distortion-rate function $D_\infty^X(R)$ is also defined in the same manner, except that the mutual-information is fixed and the distortion is what is infimized.

D. Outline

Section II considers lossy source coding for unstable Markov processes with the disturbance W_t constrained to have bounded support. A fixed-rate code at a rate arbitrarily close to $R(d)$ is constructed by encoding process into two simultaneous fixed-rate bit streams. The first stream has a rate arbitrarily close to $\log_2 A$ and encodes what is needed from the past to understand the future. It captures the information that is accumulating within the unstable process. The other stream captures those aspects of the past that are not relevant to the future and so captures the purely historical aspects of the unstable process in a way that meets the average distortion constraint. This second stream can be made to have a rate arbitrarily close to $R(d) - \log_2 A$. This historical information is examined more carefully by looking at the process going backward in time. The $R(d)$ curve for the unstable process is shown to have a shape that is bounded by the stable historical part translated by $\log_2 A$ to account for the unstable accumulation of information.

Section III reviews the delay-sensitive notion of anytime reliability and the fact that random codes exist achieving this sense of reliability over noisy channels even without any feedback. For η-difference-distortion measures, an anytime reliability $> \eta \log_2 A$ is then shown to be sufficient to encode the first bitstream across a noisy channel. The second bitstream is shown to only require classical Shannon ϵ-reliability. This completes the reduction of the lossy-estimation problem to a two-tiered reliable bit-transportation problem.

In Section IV, the problem of anytime-reliable bit-transport is directly reduced to the problem of lossy-estimation for the accumulation process using the ideas in [5], reinterpreted as information-embedding. This shows that the higher QoS requirements for the first stream are fundamental to these processes. A second stream of data is embedded into the historical segments of the unstable process and is recovered in

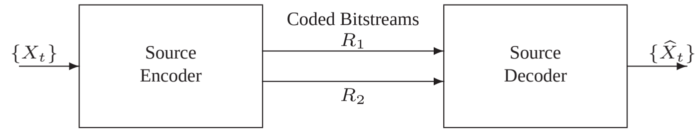

Fig. 2. The source-coding problem of translating the source into two simultaneous bitstreams of fixed rates R_1 and R_2. End-to-end delay is permitted but must remain bounded. The goal is to get $R_1 \approx \log_2 A$ and $R_2 \approx R(d) - \log_2 A$.

the classical Shannon ϵ-reliable sense. *Exponentially unstable Markov processes are thus the first nontrivial examples of stochastic processes that naturally generate two qualitatively distinct kinds of information.*

In Section V, the results are then extended to cover the Gaussian-process case with the usual squared-error distortion and a numerical example is discussed to illustrate the ideas. Although the proofs are given in terms of Gaussian processes and squared error, the results actually generalize to any driving noise distribution that has at least an exponentially decaying tail.

This paper focuses throughout on scalar Markov processes. It is possible to extend all the arguments to cover the general ARMA case[4]. The techniques used to cover the ARMA case are discussed in the control context in [6] where the state-space formulation is used. A brief discussion of how to apply those techniques is present here in Section VI.

II. Two stream source encoding: approaching $R(d)$

This section focuses on the source coding problem illustrated in Fig. 2. The goal is to transport the unstable source of (1) to the destination using a low rate while maintaining the target fidelity level. A large end-to-end delay is permitted, but it must remain bounded even as time goes to infinity. The main result of this section is:

Theorem 2.1: Given an unstable ($A > 1$) scalar Markov process as given by (1) driven by independent noise $\{W_t\}_{t\geq 0}$ with bounded support, it is possible to encode the process to average fidelity $E[|X_i - \widehat{X}_i|^\eta]$ arbitrarily close to d using two fixed-rate bitstreams assuming that both encoder and decoder have access to common randomness. By choosing a sufficiently large end-to-end delay, the first stream can be made to have rate R_1 arbitrarily close to $\log_2 A$ while the second can have rate R_2 arbitrarily close to $R^X_\infty(d) - \log_2 A$.

A. Proof strategy

The code for proving Theorem 2.1 is illustrated in Fig. 3.

- Look at time in blocks of size n and encode the values of endpoints (X_{kn-1}, X_{kn}) recursively to very high precision using rate $n(\log_2 A + \epsilon_1)$ per pair. Each block $X_{kn}, X_{kn+1}, \ldots, X_{(k+1)n-1}$ will have encoded checkpoints $(\check{X}_{kn}, \check{X}_{kn+n-1})$ at both ends.
- Use the encoded checkpoints $\{\check{X}_{kn}\}$ at the start of the blocks to transform the process in between (the history) so that it looks like an iid sequence of finite horizon problems $\vec{X}$.
- Use the checkpoints $\{\check{X}_{kn+n-1}\}$ at the end of the blocks to encode the history to fidelity d at a rate of $n(R^X_\infty(d) - \log_2 A + \epsilon_2 + o(1))$ per block.
- "Stationarize" the encoding by choosing a random starting offset so that no times t are *a priori* more vulnerable to distortion.

The source decoding proceeds in the reverse manner and gets both checkpoints and history. The two are recombined to give a reconstruction of the original source to the desired fidelity. The above strategy follows the spirit of Berger's encoding[11]. In Berger's code for the Wiener process, the first stream's rate

[4]ARMA: Autoregressive process driven by the moving average of a white process. The unstable poles give rise to streams requiring higher QoS while the residual stream can do with less. The total rate will govern performance.

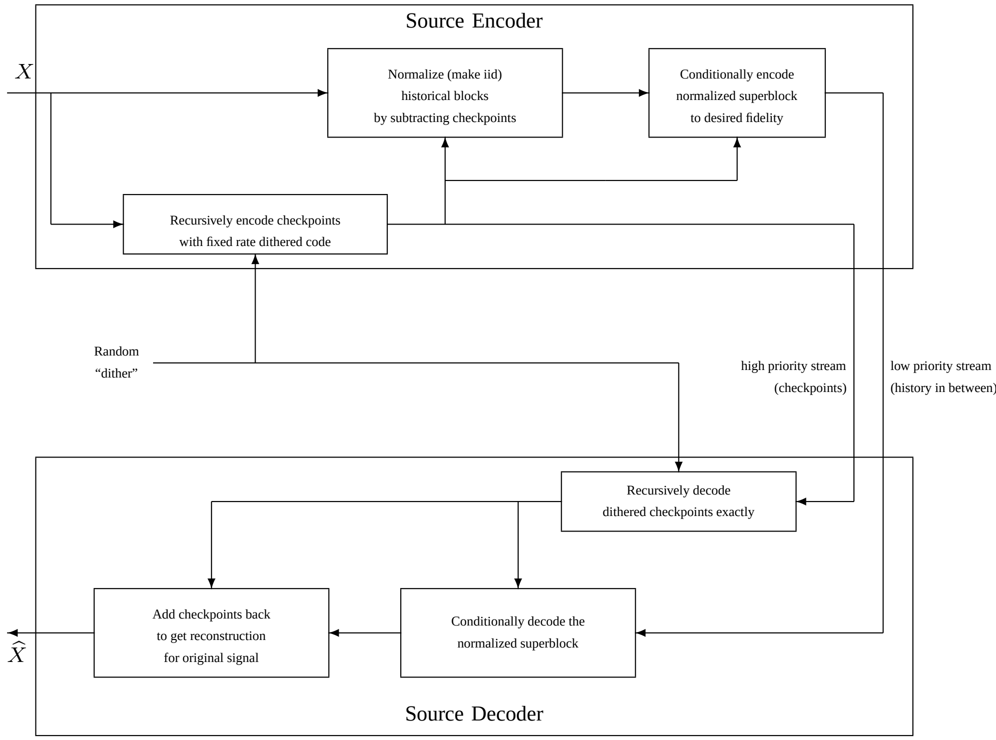

Fig. 3. A flowchart showing how to do two-stream fixed rate source coding for Markov sources and how the streams are decoded.

is negligible relative to the second stream. In our case, the first stream's rate is significant and cannot be averaged away by using large blocks n.

The detailed constructions and proof for this theorem are in the next few subsections.

B. Recursively encoding checkpoints

This section relies[5] upon the assumption of driving noise with bounded support $|W_t| \leq \frac{\Omega}{2}$, but does not care about any other property of the $\{W_t\}_{t\geq 0}$ like independence or stationarity. The distortion measure is also not important for this section.

Proposition 2.1: Given an unstable ($A > 1$) scalar Markov process as given by (1) driven by noise $\{W_t\}_{t\geq 0}$ with bounded support, it is possible to causally and recursively encode checkpoints spaced by n to arbitrarily high fidelity ($|\check{X}_{kn} - X_{kn}| \leq \Delta$, for any $\Delta > 0$ we choose) with rate R_1 arbitrarily close to $\log_2 A$ by choosing n large enough. Furthermore, if an iid sequence of independent pairs of uniform random variables $\{\Theta_i, \Theta'_i\}_{i\geq 0}$ for dithering is available to both encoder and decoder, the errors $(\check{X}_{kn-1} - X_{kn-1}, \check{X}_{kn} - X_{kn})$ can be made an iid sequence of pairs of independent uniform random variables.

Proof: First, we consider the initial condition at X_0. It can be quantized to be within an interval of size Δ by using $\log_2 \lceil \frac{\Omega_0}{\Delta} \rceil$ bits.

With a block length of n, the successive endpoints are related by:

$$X_{(k+1)n} = A^n X_{kn} + [A^{n-1} \sum_{i=0}^{n-1} A^{-i} W_{kn+i}] \tag{4}$$

The second term $[\cdots]$ on the left of (4) can be denoted $\tilde{W}_k$ and bounded by:

[5] This assumption is relaxed when we consider the Gaussian case in Section V.

$$|\tilde{W}_k| = |A^{n-1}\sum_{i=0}^{n-1} A^{-i}W_{kn+i}| \leq |A^{n-1}|\sum_{i=0}^{n-1} A^{-i}\frac{\Omega}{2} < A^n\frac{\Omega}{2(A-1)} \tag{5}$$

Proceed by induction. Assume that we have $\check{X}_{kn}$ so that $|X_{kn} - \check{X}_{kn}| \leq \frac{\Delta}{2}$ for some Δ chosen small. This clearly holds for $k = 0$. Without any further information, we know that $X_{(k+1)n}$ must lie within an interval of size $A^n\Delta + A^n\frac{\Omega}{A-1}$. By using nR_1' bits[6] to encode where the true value lies, the uncertainty is cut by a factor of $2^{nR_1'}$. To have the resulting interval of size Δ or smaller again, we must have:

$$\Delta \geq 2^{-nR_1'}A^n(\Delta + \frac{\Omega}{(A-1)}).$$

Dividing through by $\Delta 2^{-nR_1'}A^n$ and taking logarithms gives:

$$n(R_1' - \log_2 A) \geq \log_2(1 + \frac{\Omega}{\Delta(A-1)})$$

Encoding $\check{X}_{kn-1}$ given $\check{X}_{kn}$ requires very little additional rate since $|X_{kn-1} - \check{X}_{kn}| < \Omega + \Delta$ and so $\log_2\lceil\frac{\Omega}{\Delta}+1\rceil < \log_2(2+\frac{\Omega}{\Delta})$ additional bits are good enough to encode both checkpoints. Putting everything together in terms of the original R_1 gives:

$$R_1 \geq \max\left(\log_2 A + \frac{\log_2(1+\frac{\Omega}{\Delta(A-1)}) + \log_2(2+\frac{\Omega}{\Delta})}{n}, \frac{\log_2\lceil\frac{\Omega_0}{\Delta}\rceil}{n}\right) \tag{6}$$

It is clear from (6) that no matter how small a Δ we choose, by picking an n large enough the rate can get as close to $\log_2 A$ as desired. In particular, picking $n = K(\log_2\frac{1}{\Delta})^2$ works with large K and small Δ.

To get the uniform nature of the final error $\check{X}_{kn} - X_{kn}$, subtractive dithering can be used[20]. This is accomplished by adding a small iid random variable Θ_k, uniform on $[-\frac{\Delta}{2}, +\frac{\Delta}{2}]$, to the X_{kn}, and only then quantizing $(X_{kn} + \Theta_k)$ to resolution Δ. At the decoder, Θ_k is subtracted from the result to get $\check{X}_{kn}$. Similarly for $\check{X}_{kn-1}$. This results in the checkpoint error sequence $(X_{kn-1} - \check{X}_{kn-1}, X_{kn} - \check{X}_{kn})$ being iid uniform pairs over $[-\frac{\Delta}{2}, +\frac{\Delta}{2}]$, that are also independent of all the W_t and initial condition X_0. $\square$

In what follows, we always assume that Δ is chosen to be of high fidelity relative to the target distortion d (e.g. For squared-error distortion, this means that $\Delta^2 \ll d$.) as well as small relative to the the initial condition so $\Delta \ll \Omega_0$.

C. *Transforming and encoding the history*

Having dealt with the endpoints, focus attention on the historical information between them. Here, the bounded support assumption is not needed for the $\{W_t\}$, but the iid assumption is important. First, the encoded checkpoints are used to transform the historical information so that each historical segment looks iid. Then, it is shown that these segments can be encoded to the appropriate fidelity and rate when the decoder has access to the encoded checkpoints as side information.

1) Forward transformation: The simplest transformation is to effectively restart the process at every checkpoint and view time going forward. This can be considered normalizing each of the historical segments $X_{kn}^{(k+1)n-1}$ to $(\tilde{X}_{(k,i)}, 0 \leq i \leq n-1)$ for $k = 0, 1, 2, \ldots$.

$$\tilde{X}_{(k,i)} = X_{kn+i} - A^i\check{X}_{kn} \tag{7}$$

For each k, the block $\tilde{X}_k = \{\tilde{X}_{(k,i)}\}_{0\leq i\leq n-1}$ satisfies $\tilde{X}_{(k,i+1)} = A\tilde{X}_{(k,i)} + W_{(k,i)}$. By dithered quantization, the initial condition ($i = 0$) of each block is a uniform random variable of support Δ that is independent of all other random variables in the system. The initial conditions are iid across the different

[6] We can choose R_1' in such a way as to guarantee us an integer nR_1'

k. Thus, except for the initial condition, the blocks $\tilde{X}_k$ are identically distributed to the finite horizon versions of the problem.

Since $\Delta < \Omega_0$, each $\tilde{X}_k$ block starts with a tighter initial condition than the original X process did. Since the distortion measure ρ depends only on the difference, starting with a smaller initial condition while everything else is the same implies that the process requires no more bits per symbol to achieve a distortion d than did the original process. Thus:

$$R_n^{\tilde{X}}(d) \leq R_n^X(d)$$

for all n and d. So in the limit of small Δ and large n

$$\lim_{\Delta \downarrow 0} R_\infty^{\tilde{X}}(d) \leq R_\infty^X(d) \tag{8}$$

Since quantizing the initial condition to any desired precision takes only a finite number of additional bits, these bits can be amortized away as $n \to \infty$. Thus (8) can be tightened to

$$\lim_{\Delta \downarrow 0} R_\infty^{\tilde{X}}(d) = R_\infty^X(d) \tag{9}$$

In simple terms, the normalized history behaves like the finite horizon version of the problem when n is large.

2) Conditional encoding: The idea is to encode the normalized history between two checkpoints conditioned on the ending checkpoint. The decoder has access to the exact values of these checkpoints through the first bitstream and so from the point of view of coding the historical blocks, the checkpoints represent side information available at both the encoder and decoder.

For a given k, shift the encoded ending checkpoint $\check{X}_{(k+1)n-1}$ to

$$Z_k^q = \check{X}_{(k+1)n-1} - A^{n-1}\check{X}_{kn} \tag{10}$$

Z_k^q is clearly available at both the encoder and the decoder since it only depends on the encoded checkpoints. Furthermore, it is clear that

$$\tilde{X}_{(k,n-1)} - Z_k^q = (X_{(k+1)n-1} - A^{n-1}\check{X}_{kn}) - (\check{X}_{(k+1)n-1} - A^{n-1}\check{X}_{kn}) = X_{(k+1)n-1} - \check{X}_{(k+1)n-1}$$

which is a uniform random variable on $[-\frac{\Delta}{2}, +\frac{\Delta}{2}]$.

To see what rate is required for the second stream, define the conditional rate-distortion function $R_\infty^{X|Z^q,\Theta}(d)$ for the limit of long historical blocks $\tilde{X}_0^{n-1}$ conditioned on their quantized endpoint:

$$R_\infty^{X|Z^q,\Theta}(d) = \liminf_{n\to\infty} \frac{1}{n} \inf_{\{\mathcal{P}(Y_0^{n-1}|\tilde{X}_0^{n-1}, Z^q, \Theta) : \frac{1}{n}\sum_{i=0}^{n-1} E\left[|\tilde{X}_i - Y_i|^\eta\right] \leq d\}} \frac{1}{n} I(\tilde{X}_0^{n-1}; Y_0^{n-1} | Z^q, \Theta)$$

Proposition 2.2: Given an unstable ($A > 1$) scalar Markov process $\{\tilde{X}_t\}$ given by (1) together with its encoded endpoint Z^q obtained by Θ-dithered quantization to within a a uniform random variable with small support Δ, the conditional rate distortion function $R_\infty^{X|Z^q,\Theta}(d) \leq R_\infty^X(d) - \log_2 A$

Proof: From (8),(3), and (2) we know if Δ is small enough and n is large enough, that there exists a random vector Y_0^{n-1} so that $\frac{1}{n}\sum_{i=0}^{n-1} \rho(\tilde{X}_i, Y_i) = d + \epsilon_3$ and $I(\tilde{X}_0^{n-1}; Y_0^{n-1}) = n(R_\infty^X(d) + \epsilon_2)$. Decompose the relevant mutual information as:

$$I(\tilde{X}_0^{n-1}; Y_0^{n-1} | Z^q, \Theta) = -I(\tilde{X}_0^{n-1}; Z^q | \Theta) + I(\tilde{X}_0^{n-1}; Y_0^{n-1}, Z^q | \Theta) \tag{11}$$

To upper bound the conditional mutual information, we lower bound $I(\tilde{X}_0^{n-1}; Z^q|\Theta)$ and upper bound $I(\tilde{X}_0^{n-1}; Y_0^{n-1}, Z^q|\Theta)$. The first term is easily lower bounded by $\lfloor n \log_2 A \rfloor$ for Δ small enough since:

$$\begin{aligned} I(\tilde{X}_0^{n-1}; Z^q|\Theta) &= H(Z^q|\Theta) - H(Z^q|\tilde{X}_1^n, \Theta) \\ &= H(Z^q|\Theta) \\ &\geq H(Z^q|\Theta, W_0^{n-2}) \\ &\geq \lfloor \log_2 A^{n-1} \rfloor \\ &= \lfloor (n-1) \log_2 A \rfloor \end{aligned} \tag{12}$$

since conditioned on the final dither Θ, the quantized endpoint is a discrete random variable that is a deterministic function of $\tilde{X}_{n-1}$ and conditioning reduces entropy. But Z^q conditioned on the driving noise W_0^{n-2} is just the Δ-precision quantization of A^{n-1} times a uniform random variable of width Δ and hence has discrete entropy $\geq \log_2 A^{n-1}$.

The second term of (11) is a little more subtle to bound. We need to establish

$$I(\tilde{X}_0^{n-1}; Y_0^{n-1}, Z^q|\Theta) \leq n(R_\infty^X(d) + \epsilon_2) + o(n) \tag{13}$$

Expand the mutual information as:

$$\begin{aligned} I(\tilde{X}_0^{n-1}; Y_0^{n-1}, Z^q|\Theta) &= I(\tilde{X}_0^{n-1}; Y_0^{n-1}|\Theta) + I(\tilde{X}_0^{n-1}; Z^q|\Theta, Y_0^{n-1}) \\ &= I(\tilde{X}_0^{n-1}; Y_0^{n-1}|\Theta) + H(Z^q|\Theta, Y_0^{n-1}) - H(Z^q|\Theta, Y_0^{n-1}, \tilde{X}_0^{n-1}) \\ &\leq I(\tilde{X}_0^{n-1}; Y_0^{n-1}|\Theta) + H(Z^q|\Theta, Y_{n-1}) \\ &= n(R_\infty^X(d) + \epsilon_2) + H(Z^q - Q_{(\Delta,\Theta)}(Y_{n-1})|\Theta, Y_{n-1}) \\ &\leq n(R_\infty^X(d) + \epsilon_2) + H(Z^q - Q_{(\Delta,\Theta)}(Y_{n-1})|\Theta) \end{aligned}$$

The first inequality comes from dropping a negative term[7] and dropping the conditioning on Y_0^{n-2}. $Q_{(\Delta,\Theta)}$ is used to denote the dithered scalar quantizer used to generate the encoded checkpoints, just appropriately translated so it can apply to the $\tilde{X}$ giving $Z^q = Q_{(\Delta,\Theta)}(\tilde{X}_{n-1})$. As such, $Q_{(\Delta,\Theta)}$ can be applied to Y_{n-1} so that $Z^q - Q_{(\Delta,\Theta)}(Y_{n-1}) = S\Delta$ where S is an integer-valued random variable representing how many steps up or down the Δ-quantization ladder are needed to get from $Q_{(\Delta,\Theta)}(Y_{n-1})$ to Z^q. This tells us:

$$I(\tilde{X}_0^{n-1}; Y_0^{n-1}, Z^q|\Theta) \leq n(R_\infty^X(d) + \epsilon_2) + H(S)$$

To bound the entropy in S, observe that $|S| \leq 1 + \frac{|\tilde{X}_{n-1} - Y_{n-1}|}{\Delta}$ since the quantized points are no more

[7]Since Z^q is discrete once conditioned on the dither Θ, H is the regular discrete entropy here.

than $\frac{\Delta}{2}$ from the originals. The η-th moment of S can be bounded:

$$\begin{aligned}
E[|S|^\eta] &\leq E[(1 + \frac{|\tilde{X}_{n-1} - Y_{n-1}|}{\Delta})^\eta] \\
&\leq E[(2\max\left(1, \frac{|\tilde{X}_{n-1} - Y_{n-1}|}{\Delta}\right))^\eta] \\
&= E[2^\eta \max\left(1^\eta, (\frac{|\tilde{X}_{n-1} - Y_{n-1}|}{\Delta})^\eta\right)] \\
&\leq E[2^\eta + 2^\eta \frac{|\tilde{X}_{n-1} - Y_{n-1}|^\eta}{\Delta^\eta}] \\
&= 2^\eta + \frac{2^\eta}{\Delta^\eta} E[|\tilde{X}_{n-1} - Y_{n-1}|^\eta] \\
&\leq 2^\eta + \frac{2^\eta}{\Delta^\eta} E[\sum_{i=0}^{n-1} |\tilde{X}_i - Y_i|^\eta] \\
&\leq 2^\eta + \frac{2^\eta}{\Delta^\eta} n(d + \epsilon_3)
\end{aligned}$$

Notice that for large enough n and/or small enough Δ, only the $O(n)$ term matters. For simplicity, we assume that n is large enough or Δ is small enough so that: $E[|S|^\eta] \leq 2^{\eta+1} n \frac{(d+\epsilon_3)}{\Delta^\eta}$

Applying the Markov inequality gives us:

$$\mathcal{P}(|S| \geq s) \leq \min(1, \frac{2^{\eta+1} n(d + \epsilon_3)}{\Delta^\eta} s^{-\eta}) \tag{14}$$

Since an integer S can be encoded into bits using a self-punctuated code[8] using less than $3 + 2\log_2(|S|)$ bits to encode $S \neq 0$, the entropy of S must be bounded as follows:

$$\begin{aligned}
H(S) &\leq 3 + 2E[\log_2(|S|)] \\
&= 3 + 2\int_0^\infty \mathcal{P}(\log_2(|S|) > l) dl \\
&= 3 + 2\int_0^\infty \mathcal{P}(|S| > 2^l) dl \\
&\leq 3 + 2\int_0^\infty \min(1, \frac{2^{\eta+1} n(d + \epsilon_3)}{\Delta^\eta} 2^{-\eta l}) dl \\
&= 3 + \frac{2}{\eta}\log_2(\frac{2^{\eta+1} n(d + \epsilon_3)}{\Delta^\eta}) + 2\int_0^\infty 2^{-\eta u} dl \\
&= \frac{2}{\eta}\log_2 n + \log_2 \frac{1}{\Delta^2} + (5 + \frac{2}{\eta}(1 + \log_2(d + \epsilon_3))) + \frac{2}{\eta \ln 2}
\end{aligned}$$

The $\frac{2}{\eta}\log_2 n$ term is certainly $o(n)$. The only other term that might raise concern is $\log_2 \frac{1}{\Delta^2}$, but that is $o(n)$ since (6) tells us that we are already required to choose n much larger than that to have R_1 close to $\log_2 A$ in the first stream.

With (13) established, we can apply it along with (12) to (11) giving us:

$$I(\tilde{X}_0^{n-1}; Y_0^{n-1} | Z^q, \Theta) \leq n(R_\infty^X(d) - \log_2 A + \epsilon_2) + o(n) \tag{15}$$

Taking n to ∞ and dividing through by n establishes the desired result. □

[8] First encode the sign of S using a single bit, then give the binary expansion of $|S|$ with each digit followed by a 0 if it is not the last digit, and a 1 if it is the last digit.

This conditional-rate-distortion function in Proposition 2.2 has a corresponding coding theorem:

Proposition 2.3: Given an unstable ($A > 1$) scalar Markov process $\{\tilde{X}_t\}$ given by (1) together with its n-spaced pairs of encoded checkpoints $\{\check{X}\}$ obtained by dithered quantization to within iid uniform random variables with small support Δ, for every $\epsilon_4 > 0$, there exists an M large enough so that a conditional source-code exists that maps a length M superblock of the historical information $\{\tilde{X}_k\}_{0 \leq k < M}$ into a superblock of $\{T_k\}_{0 \leq k < M}$ satisfying:

$$\frac{1}{M}\sum_{k=0}^{M-1}\frac{1}{n}\sum_{j=1}^{n} E[\rho(\tilde{X}_{(k,j)}, T_{(k,j)})] \leq d + \epsilon_4$$

By choosing n large enough, the rate of the superblock code can be made as close as desired to $R^X_\infty(d) - \log_2 A$ if the decoder is also assumed to have access to the encoded checkpoints $\check{X}_{kn}$.

Proof: The idea is to encode M of the $\tilde{X}_k$ blocks together and use conditioning on the information in the M encoded checkpoints at the end of each. The $\tilde{X}_k$ vector and Z^q_k pair have a joint distribution, but are iid across k by the independence properties of the subtractive dither and the $W_{(k,i)}$. Furthermore, the $\tilde{X}_{(k,i)}$ are bounded and as a result, the all zero reconstruction results in a bounded distortion on the $\tilde{X}$ vector. Even without the bounded support assumption, Theorem 2.2 reveals that there is a reconstruction based on the Z^q_k alone that has bounded average distortion where the bound does not even depend on n.

Since the side information Z^q_k is available at both encoder and decoder, Proposition 2.2 and the classical conditional rate distortion coding theorems[21] tell us that there exists a block-length $M(n)$ so that codes exist which satisfy the properties required as we let n get large. □

D. Putting history together with checkpoints

All that remains is showing how the decoder can combine the two streams to get the desired rate/distortion performance.

The rate side is immediately obvious since there is $\log_2 A$ from Proposition 2.1 and $R^X_\infty(d) - \log_2 A$ from Proposition 2.3. The sum is as close to $R^X_\infty(d)$ as desired. On the distortion side, the decoder runs (7) in reverse to get reconstructions. Suppose that $T_{(k,i)}$ are the encoded transformed source symbols from the code in Proposition 2.3. Then $\widehat{X}_{kn+i} = T_{(k,i)} + A^i \check{X}_{kn}$ and so $X_{kn+i} - \widehat{X}_{kn+i} = \tilde{X}_{(k,i)} - T_{(k,i)}$. Since the differences are the same, so is the average distortion and any other property that we had for the vector quantization problem with known side information.

E. "Stationarizing" the code

The underlying process is non-stationary so there is no hope to make the encoding truly stationary. However, as it stands, only the average distortion across each of the Mn length superblocks is close to d in expectation giving the resulting code a "cyclostationary" performance. Nothing guarantees that source symbols at every time will have the same level of expected fidelity. To fix this, a standard trick can be applied by making the encoding have two phases:

- A first phase that lasts for a random T time-steps. T is a random integer chosen uniformly from $0, 1, \ldots Mn-1$ based on common randomness available to the encoder and decoder. During the first phase, all source symbols are encoded to fidelity Δ recursively.
- A second phase that applies the two-part code described here but starts at time $T+1$.

The extra rate required in the first phase is negligible on average since it is a one-time cost. It just translates into an extra delay as this first phase message has to drain through the rate $R_1 > \log_2 A$ data stream while the second phase of the first bitstream can be encoded in a causal fashion with only a delay of n. The rest of the end-to-end delay is determined by the total length Mn of the superblock chosen inside Proposition 2.3.

Because the first phase is guaranteed to be high fidelity and all other time positions are randomly and uniformly assigned positions within the superblock of size M, the expected distortion $E[|X_i - \widehat{X}_i|^\eta] \leq d + \epsilon_4$ for every i. The code actually does better than that since the probability of excess average distortion over a long block is also guaranteed to go to zero.[9]

This proves Theorem 2.1.

F. Time-reversal and the essential phase transition

With Theorem 2.1 proved, it is useful to make some observations. First, the distortion performance of the code is entirely based on the conditional rate-distortion curve for the historical segments. The checkpoints merely contribute a $\log_2 A$ term in the rate. Moreover, since the above code can approach the $R_\infty^X(d)$ bound as closely as desired, this means that Proposition 2.2 must in fact hold with equality:

$$R_\infty^{X|Z^q,\Theta}(d) = R_\infty^X(d) - \log_2 A \tag{16}$$

The nature of historical information in the unstable Markov process described by (1) can be explored more fully by transforming the historical blocks going locally backward in time. The informational distinction between the process going forward and the purely historical information parallels the concepts of information production and dissipation explored in the context of the Kalman Filter [15]. One interesting consequence of this is a qualitative difference between the noncausal $D(R)$ and the causal $D_{\textsf{seq}}(R)$ defined in [22] for the unstable process. While $D_{\textsf{seq}}(R)$ goes to infinity as the rate approaches $R = \log_2 A$ from the right, $D(R)$ instead approaches a finite limit!

1) Backwards process: Call X^- the "backwards in time process" related to the original process defined in (1).

$$X_t^- = A^{-1} X_{t+1}^- - A^{-1} W_t^- \tag{17}$$

where the $\{W_t^-\}$ are iid with the same distribution as $\{W_t\}$ and the process is initialized to either $X_0^- = 0$ or its steady-state distribution. Time runs in the negative direction. Since $|A^{-1}| < 1$, this is a stable Markov process and falls under the classical theorems of [9].

2) Bounding the forward process: The historical blocks, conditioned on their endpoints, are essentially stable in nature since they can be looked at going backward. This allows a simplification in the code depicted in Fig. 3: the encoding of the historical information can be done unconditionally and on a block-by-block basis.

Theorem 2.2: The rate-distortion function for the unstable Markov process is bounded above by $\log_2 A$ plus the the required rate to encode a backwards-in-time version X^- of the process starting with a known endpoint.

$$R_\infty^X(d) \leq \log_2 A + R_\infty^{X^-}(d) \tag{18}$$

or expressed in terms of distortion-rate functions for $R > \log_2 A$:

$$D_\infty^X(R) \leq D_\infty^{X^-}(R - \log_2 A)$$

This further implies that the process undergoes a phase transition from infinite to bounded distortions at the rate $\log_2 A$.

Proof: Since (16) is known, all that needs to be shown is that

$$R_\infty^{X|Z^q,\Theta}(d) \leq R_\infty^{X^-}(d) \tag{19}$$

When looked at going backwards in time[10], the untransformed k-th historical block satisfies the equation

$$X_{kn+i} = A^{-1} X_{kn+i+1} - A^{-1} W_{kn+i} \tag{20}$$

[9] This property is inherited from the repeated use of independent conditional rate-distortion codes in the second stream.

[10] Time is not being reversed for the entire process, just over a fixed size block with known endpoints. Consequently, the technical complications that can occur when reversing time in the infinite horizon case do not come up here.

for every $0 \leq i \leq n$ with endpoints X_{kn} and $X_{(k+1)n}$ known to each be within $\frac{\Delta}{2}$ of $\check{X}_{kn}$ and $\check{X}_{(k+1)n}$ respectively. This is identical to (17) except that the known final condition is large rather than small. This can be easily remedied by a reversible transformation in the style of (7) except starting with the dithered endpoint $\check{X}_{(k+1)n}$. Define $\check{X}_{(k,i)}$ by:

$$\check{X}_{(k,i)} = X_{kn+i} - A^{i-n}\check{X}_{(k+1)n} \tag{21}$$

When viewed going backwards, this has an "initial" condition of $\check{X}_{(k,n)} = X_{(k+1)n} - \check{X}_{(k+1)n}$ being a uniform random variable on $[-\frac{\Delta}{2}, +\frac{\Delta}{2}]$ that is independent of all the W by the properties of dithered quantization. This evolves backwards according to

$$\check{X}_{(k,i)} = A^{-1}\check{X}_{k,i+1} - A^{-1}W_{k,i} \tag{22}$$

and ends at

$$\begin{aligned}\check{X}_{(k,0)} &= X_{kn} - A^{-n}\check{X}_{(k+1)n} \\ &= (X_{kn} - \check{X}_{kn}) + (\check{X}_{kn} - A^{-n}\check{X}_{(k+1)n}) \\ &= (X_{kn} - \check{X}_{kn}) - A^{-n}Z_k^q\end{aligned}$$

where Z_k^q is from (10) and is calculated from the two quantized boundary conditions. $-A^{-n}Z_k^q$ is a value within Δ of $-\sum_{j=1}^{n} A^{-j}W_{k,j-1})$. When n is large, the $\check{X}$ vectors behave close to the steady-state behavior of X^-. By the properties of dithered quantization, $\check{X}_{(k,0)}$ is thus known to be somewhere uniformly distributed on $[-A^{-n}Z_k^q - \frac{\Delta}{2}, -A^{-n}Z_k^q + \frac{\Delta}{2}]$. The mapping from X to $\check{X}$ is invertible conditioned on the side information provided by the quantized endpoints.

The second stream can be encoded to distortion $d + \epsilon$ by simply ignoring the boundary conditions for $\check{X}$ and taking n large enough. Thus:

$$R_{\infty}^{X|Z^q,\Theta}(d) \leq R_{\infty}^{X^-}(d)$$

At rates less than $\log_2 A$, the distortion for the original X process is necessarily[11] infinite. The reverse process X^- is much better behaved since it is bounded $|X_t^-| \leq \frac{\Omega}{A-1}$ and hence has finite difference distortion for all $\eta > 0$ even at zero rate. Thus, the $D_{\infty}^{X}(R)$ must transition from infinite to bounded distortions at $R = \log_2 A$. □

Notice that there are no explicitly infinite distortions in the original setup of the problem. Consequently, the appearance of infinite distortions is interesting as is the abrupt transition from infinite to finite distortions at the critical rate of $\log_2 A$. Theorem 2.2 tells us that the unstable $A > 1$ Markov processes are nonclassical only as they evolve into the future. The historical information is no worse than a stable Markov processes that fleshes out the unstable skeleton of the process.

III. Quality of service requirements for communicating unstable processes: Sufficiency

In a very real sense, the first stream in Theorem 2.1 represents an initial description of the process to some fidelity, while the second represents a refinement of the description [23]. Theorem 2.1 points to the fact that asymptotically such processes are successively refinable for the particular choice of $R_1 \approx \log_2 A$. These two descriptions turn out to be qualitatively different when it comes to transporting them across a noisy channel.

In Section III-A, the sense of anytime reliability is reviewed and related to classical results on sequential coding for noisy channels. Then in Section III-B, anytime reliable communication is shown to be sufficient for protecting the encoding of the checkpoint process. Finally in Section III-C, it is shown that it is sufficient to transport the historical information using traditional Shannon ϵ-reliability.

[11] This will be established rigorously in Theorem 4.2 where finite distortion implies the ability to carry $\approx \log_2 A$ bits through the communication medium.

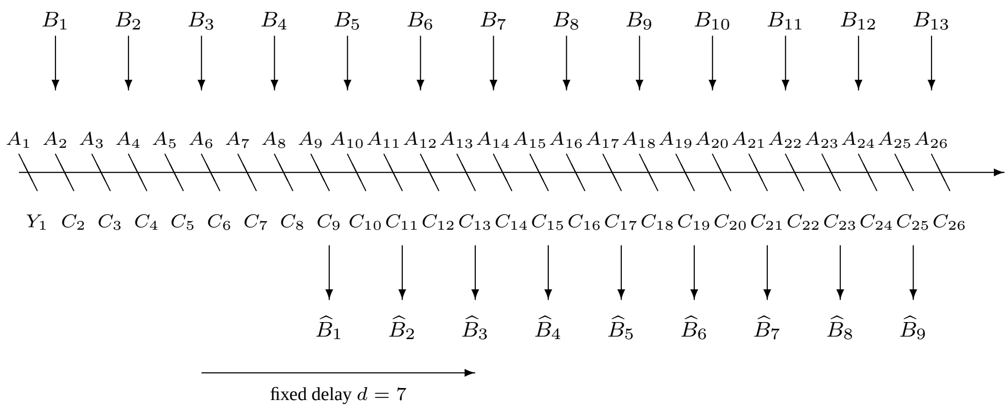

Fig. 4. The timeline in a rate $\frac{1}{2}$ delay 7 code. Both the encoder and decoder must be causal so X_i and $\widehat{B}_i$ are functions only of quantities to the left of them on the timeline. If noiseless feedback is available, the X_i can also have an explicit functional dependence on the C_1^{i-1} that lie to the left on the timeline.

A. *Anytime reliability*

It should be clear that the encoding given for the checkpoint process in Section II-B is very sensitive to bit errors since it is decoded recursively in a way that will propagate errors. To specify the quality of service requirement, we need to look more closely at the times at which individual bits and their reconstructions occur relative to the channel uses. (Illustrated in Fig. 4)

Definition 3.1: A *discrete time channel* is a probabilistic system with an input. At every time step t, it takes an input $a_t \in \mathcal{A}$ and produces an output $c_t \in \mathcal{C}$ with probability[12] $p(C_t|a_1^t, c_1^{t-1})$ where the notation a_1^t is shorthand for the sequence $a_1, a_2, \ldots, a_t$. In general, the current channel output is allowed to depend on all inputs so far as well as on past outputs.

The channel is *memoryless* if conditioned on a_t, C_t is independent of any other random variable in the system that occurs at time t or earlier and all that needs to be specified is $p_t(C_t|a_t)$. The channel memoryless and stationary if $p_t(C_t|a_t) = p(C_t|a_t)$ for all times t.

Definition 3.2: A *rate R channel-encoder $\mathcal{E}$ without feedback* is a sequence of maps $\{\mathcal{E}_t\}$. The range of each map is the set $\mathcal{A}$. The t-th map takes as input the available data bits $B_1^{\lfloor Rt \rfloor}$.

Randomized encoders also have access to a continuous uniform[13] random variable denoting the common randomness available in the system.

Definition 3.3: A *delay ϕ rate R channel-decoder* is a sequence of maps $\{\mathcal{D}_i\}$. The range of each map is just an estimate $\widehat{B}_i$ for the i-th bit taken from $\{0, 1\}$. The i-th map takes as input the available channel outputs $C_1^{\lceil \frac{i}{R} \rceil + \phi}$ which means that it can see ϕ time units beyond the time when the desired bit first had a chance to impact the channel inputs.

Randomized decoders also have access to the continuous uniform random variables denoting common randomness.

The maximum rate achievable for a given sense of reliable communication is called the associated capacity. Shannon's classical ϵ-reliability requires that for a suitably large end-to-end delay[14] ϕ that the probability of error on each bit is below a specified ϵ. The *Shannon classical capacity* C_{Shannon} can

[12]This is a probability mass function in the case of discrete alphabets $\mathcal{C}$, but is more generally an appropriate probability measure over the output alphabet $\mathcal{C}$.

[13]Continuous uniform random variables are just used because they each represent an infinite sequence of shared random bits.

[14]Traditionally, the community has used block-length for a block code as the fundamental quantity rather than delay. It is easy to see that doing encoding and decoding in blocks of size n corresponds to a delay ϕ of between n and $2n$ on the individual bits being communicated.

also be calculated in the case of stationary memoryless channels by solving an optimization problem:

$$C_{\text{Shannon}} = \sup_{\mathcal{P}(A)} I(A;C)$$

where the maximization is over the input probability distribution and $I(A;C)$ represents the mutual information through the channel [8]. Feedback does not change C_{Shannon}. This is referred to as a single letter characterization of channel capacity for memoryless channels. Similar formulas exist using limits for channels with memory. There is another sense of reliability and its associated capacity C_0 called *zero-error capacity* which requires the probability of error to be exactly zero with sufficiently large n. It does not have a simple single-letter characterization [24] without feedback, though it can be easily computed if feedback is available.

Rather than considering the system as a bit pipe that emits individual bit estimates one at a time, suppose that the decoder produces estimates $\widehat{B}_i(t)$ that are a function of time t. In a traditional fixed-delay communications system with a delay of ϕ time units, $\widehat{B}_i(t)$ is frozen beyond the time $\frac{i}{R} + \phi$ when the decoder is forced to commit to a particular value. *However, there is no reason to impose such a fixed-delay constraint in principle.*

Consider maximum-likelihood decoding[25] or sequential-decoding [26], [27] as applied to an infinite tree code like the one illustrated in Fig. 5. The estimates $\widehat{B}_i(t)$ describe the current estimate for the most likely path through the tree based on the channel outputs received so far. Because of the possibility of "backing up," in principle the estimate for $\widehat{B}_i$ could change at any point in time. The theory of both ML and sequential decoding tells us that generically, the probability of bit error on bit i approaches zero exponentially with increasing delay. The best-path interpretation and the exponential convergence tell us that it is most convenient to look at errors on the entire prefix B_1^i.

Definition 3.4: A rate R *anytime communication system* over a noisy channel is an encoder $\mathcal{E}$ and decoder $\mathcal{D}$ pair such that:

- Data bit B_i enters the encoder at time $\frac{i}{R}$
- The encoder produces a channel input at integer times based on all information that it has seen so far.
- The decoder produces updated channel estimates $\widehat{B}_i(t)$ for all $i \leq Rt$ based on all channel outputs observed till time t

A rate R communication system achieves *anytime reliability* α if there exists a constant K such that:

$$\mathcal{P}(\widehat{B}_1^i(t) \neq B_1^i) \leq K 2^{-\alpha(t - \frac{i}{R})} \tag{23}$$

holds for every i. The probability is taken over the channel noise, the data bits B, and all of the common randomness available in the system. If (23) holds for every possible realization of the data bits B, then we say that the system achieves *uniform anytime reliability* α.

Communication systems that achieve *anytime reliability* are called *anytime codes* and similarly for *uniform anytime codes*.

"Anytime" can be considered a synonym for delay-universality. Since decoding delay plays the role for sequential codes that block-length does for block codes, asking for anytime reliability is analogous to looking at rateless block code constructions where the code achieves an exponentially decreasing probability of error regardless of when we truncate it. The key difference is that by changing the block-length, the rate of the code changes along with the end-to-end delay and error probability. Changing decoding delay alone impacts end-to-end bit error probability while holding the rate constant. The difference between uniform anytime reliability and simple anytime reliability is not that significant since it is always possible to convert an anytime code into a uniform anytime code by the addition of common randomness used to turn the input bitstream into iid fair coin tosses by XORing them with a one-time-pad.

Although it is easy to see that anytime reliability implies that the probability of error on every bit must eventually go to zero, we do not demand that we know in advance exactly when it is going to get to zero.

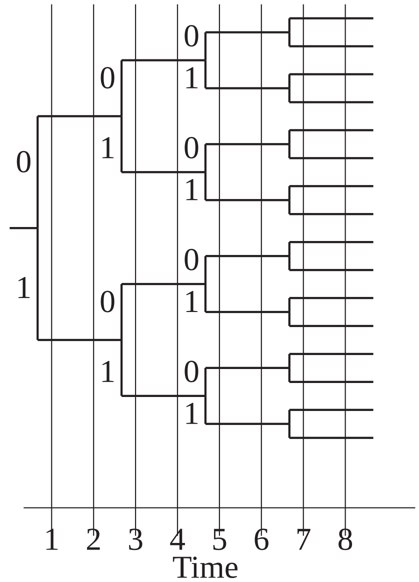

Fig. 5. A channel encoder viewed as a tree. At every integer time, each path of the tree has a channel input symbol. The path taken down the tree is determined by the data bits to be sent. Infinite trees have no intrinsic target delay and bit/path estimates can get better as time goes on.

This makes anytime reliability less demanding than Shannon's zero-error reliability [24]. The advantage is that anytime reliability is available for many channels (as Theorem 3.5 shows) that have no zero-error capacity.

In traditional analysis, random ensembles of infinite tree codes were viewed as idealizations used to study the asymptotic behavior of finite sequential encoding schemes such as convolutional codes. We can instead view the traditional analysis as telling us that random infinite tree codes achieve anytime reliability. In particular, we know from the analysis of [25], [26] that at rate R bits per channel use, we can achieve anytime reliability α equal to the block random coding error exponent. Pinsker's argument in [28] as generalized in [7] tells us also that we can not do any better, at least in the high-rate regime for symmetric channels. We summarize this interpretation in the following theorem:

Theorem 3.5: Random anytime codes exist for all DMCs For a stationary discrete memoryless channel (DMC) with capacity C, random sequential codes exist without feedback at all rates $R < C$ that have anytime reliability $\alpha = E_r(R)$ where $E_r(R)$ is the random coding error exponent as calculated in base 2.

Proof: Interpret the random ensemble of infinite tree codes as a single code with both encoder and decoder having access to the common-randomness used to generate the entire code-tree. Populate the tree with iid channel inputs drawn from the distribution that achieves $E_r(R)$ for block codes. Theorem 7 in [25] tells us that the code achieves anytime reliability $\alpha = E_r(R)$ since the analysis uses the same infinite ensemble for all i and delays.

Alternatively, this can be seen from first principles for ML decoding by observing that any false path $\tilde{B}_1^i$ can be divided into a true prefix B_1^{j-1} and a false suffix $\tilde{B}_j^i$. The iid nature of the channel inputs on the code tree tells us that the true code-suffix corresponding to the received channel outputs from time $\frac{j}{R}$ to t is independent of any false code-suffix. Since there are $\leq 2^{R(t-\frac{j}{R})}$ such false code-suffixes (ignoring integer effects) at depth j, Gallager's random block-coding analysis applies since all it requires is pairwise independence between true and false codewords.

$$\begin{aligned} & \mathcal{P}(\widehat{B}_j(t) \neq B_j | B_1^{j-1} \text{ already known}) \\ \leq \; & \mathcal{P}(\text{error on random code with } 2^{R(t-\frac{j}{R})} \text{ words and block length } t - \lceil \frac{j}{R} \rceil) \\ \leq \; & 2^{-(t-\lceil \frac{j}{R} \rceil)E_r(R)} \\ \leq \; & 2^{-(t-\frac{j}{R}-1)E_r(R)} \end{aligned}$$

The probability of error on B_1^i can be bounded by the union bound over $j = 1 \ldots i$.

$$\begin{aligned} \mathcal{P}(\widehat{B}_1^i(t) \neq B_1^i) & \leq \sum_{j=1}^{i} \mathcal{P}(\widehat{B}_j(t) \neq B_j | B_1^{j-1} \text{ already known}) \\ & \leq \sum_{j=1}^{i} 2^{-(t-\frac{j}{R}-1)E_r(R)} \\ & < \sum_{j=0}^{\infty} 2^{-(t-\frac{i}{R}-j-1)E_r(R)} \\ & = K 2^{-(t-\frac{i}{R})E_r(R)} \end{aligned}$$

The exponent for the probability of error is dominated by the shortest codeword length in the union bound, which corresponds to $t - \frac{i}{R}$. □

Since all that is required is pairwise independence, the symbols on the tree can be generated using a time-varying random convolutional code with a growing constraint length equal to the number of data bits available at the encoder [8]. Thus the common randomness required is not exponential in t, though it is still quadratic since the channel input at time t requires $O(t)$ fair coin tosses to calculate from the data bits. However, the random nature of the code ensures that the distribution of the data bits B does not matter and so we have showed uniform anytime reliability with infinite common-randomness. In [14], further arguments are given showing that deterministic anytime codes exist at the same reliabilities without having to assume infinite common randomness, although these are not uniform anytime codes since they depend on the data bits being fair coin tosses. Schulman's results on good distance properties for tree codes can be interpreted as showing that low rate deterministic uniform anytime codes exist, although not with reliabilities specified by the random coding exponent E_r [29].

The important thing to understand about anytime reliability is that it is not considered to be a proxy used to study encoder/decoder complexity[15] as traditional reliability functions[16] often are [8]. Instead, the anytime reliability parameter α indexes a sense of reliable transmission for a bitstream in which the probability of bit error tends to zero exponentially as time goes on. In the next section, this sense of reliability is shown to be helpful in communicating the checkpoint process.

[15]The question of the complexity of anytime codes is a fair one to ask. Fundamentally, the complexity required to achieve the true anytime property without feedback is infinite since both the encoder and decoder must at the minimum remember all the received symbols so far. Practically, it might be useful to use convolutional codes with very long constraint lengths so that the asymptotic probability of error is practically, if not actually, zero. Sequential decoding of such codes will then give the anytime property for moderate delay values. The loss of rate relative to capacity suffered by practical sequential decoding is not likely to be an issue in these contexts. After all, in such applications, the required anytime reliability itself will force us to back significantly away from capacity. With feedback, [7] gives constructions that do not have infinite complexity while still achieving anytime reliability.

[16]For convolutional codes, it is more popular classically to look at the exponent with respect to the constraint length rather than delay. This is because the constraint length is the natural proxy for encoder complexity. Decoder complexity is classically studied by looking at the computational requirements for sequential decoding which does not depend on the constraint length — only on how far we are from capacity [27], [26].

B. Sufficiency for the checkpoint process

The effect of any bit error in the checkpoint encoding of Section II-B will be to throw us into a wrong bin of size Δ. This bin can be at most $A^n \Delta < A^n \frac{\Omega}{A-1}$ away from the true bin. The error will then grow by a factor A^n as we move from checkpoint to checkpoint.

If we are interested in the $\eta-$difference distortion, then the error is growing by a factor of $A^{n\eta}$ per checkpoint, or a factor of A^η per unit of time. As long as the probability of error on the data bits goes down faster than that, the expected error will be small. This parallels Theorem 4.1 in [5] and results in:

Theorem 3.6: An anytime reliability $\eta \log_2 A$ is sufficient for the checkpoint stream:

Suppose that a communication system provides us with uniform anytime reliability $\alpha > \eta \log_2 A$ for the checkpoint stream at rate R_1. Then given sufficient end-to-end delay ϕ, it is possible to reconstruct the checkpoints to arbitrarily high fidelity in the η-distortion sense.

Proof: Let $\check{X}'_{kn}(\phi)$ be the best estimate of the checkpoint $\check{X}_{kn}$ at time $\frac{kn}{R_1} + \phi$. By the anytime reliability property, grouping the message bits into groups of nR_1 at a time, and the nature of exponentials, it is easy to see that there exists a constant K' so that:

$$\begin{aligned} E[|\check{X}'_{kn}(\phi) - \check{X}_{kn}|^\eta] &\leq \sum_{j=0}^{k} K' 2^{-\alpha(\phi+nj)} A^{jn\eta} \frac{\Omega}{A-1} \\ &= K'' 2^{-\alpha\phi} \sum_{j=0}^{k} 2^{-jn(\alpha+\eta\log_2 A)} \\ &\leq K'' 2^{-\alpha\phi} \sum_{j=0}^{\infty} 2^{-jn(\alpha+\eta\log_2 A)} \\ &= K''' 2^{-\alpha\phi} \end{aligned}$$

where K''' is a constant that depends on the anytime code, rate R_1, support Ω, and unstable A. Thus by choosing ϕ large enough, $2^{-\alpha\phi}$ will become small enough so that $K''' 2^{-\alpha\phi}$ is as small as we like and the checkpoints will be reconstructed to arbitrarily high fidelity. □

Theorem 3.6 applies even in the case that $A = 1$ and hence answers the question posed by Berger in [12] regarding the ability to estimate an unstable process over a noisy channel without perfect feedback. Theorem 3.5 tells us that it is in principle possible to get anytime reliability without any feedback at all. Thus we can track unstable processes without feedback, or with only noisy feedback[17].

C. QoS for the history process

It is easy to see that the history information for the two stream code does not propagate errors from superblock to superblock and so does not require any special QoS beyond what one would need for an iid or stationary-ergodic process.

Theorem 3.7: Shannon ϵ-error reliability is good enough for the history process:

Given a communication system that can transport blocks of data reliably meeting any block-error probability ϵ given a long enough block-length, then that communication system can be used to reliably transport the second (historical) information stream generated by the fixed-rate source code of Theorem 2.1 in that the expected end-to-end distortion can be made arbitrarily close to the distortion achieved by the code over a noiseless channel.

Proof: Since the impact of a bit error is felt only within the superblock, no propagation of errors needs to be considered. The time-reversal argument of Section II-F tells us that we have a maximum possible

[17] This idea of tracking an unstable process using an anytime code is used in [30], [31] used over a noisy feedback link to study the reliability functions for communication using ARQ schemes and expected delay. The sequence number of the block is considered to be an unstable process that needs to be tracked at the encoder.

distortion on the historical component since the disturbance support Ω is bounded. Thus the standard achievability argument [8] for $D(R)$ tells us that as long as the probability of block error can be made arbitrarily small with increasing block-length, then the additional expected distortion induced by bit errors will also be arbitrarily small. $\square$

The curious fact here is that the QoS requirements of the second stream of data only need to hold on a superblock-by-superblock basis. To achieve a small ensemble average distortion, there is no need to have a secondary bitstream available with error probability that gets arbitrarily small with increased delay. The secondary channel could be nonergodic and go into outage[18] for the entire semi-infinite length of time as long as that outage event occurs sufficiently rarely so that the average on each superblock is kept small.

D. The reduction of lossy compression to two-tiered bit-transport

Theorems 3.6 and 3.7 together with the source code of Theorem 2.1 combine to establish that the problem of d-lossy source coding over a noisy channel asymptotically reduces to the problem of communicating bits at rate $R(d)$ over a noisy channel, wherein a substream of bits of rate $\approx \log_2 A$ is given an anytime reliability of at least $\eta \log_2 A$. This reduction is in the sense of Section VII of [5]: any channel that is good enough to solve the second pair-of-problems is good enough to solve the first problem.

IV. Quality of service requirements for communicating unstable processes: Necessity

The classical converse to the rate-distortion theorem already assures us that we require communication resources capable of carrying at least rate $R(d)$ giving us a requirement for at least traditional Shannon ϵ-reliability. However, the goal is to show that such unstable processes require communication resources capable of supporting two-tiered service: a core of rate $\log_2 A$ with anytime-reliability of at least $\eta \log_2 A$, and the rest with Shannon reliable transport. To do this, this section proceeds in stages and follows the asymptotic equivalence approach of [5].

In Section IV-A, a pair of communication problems (the endpoint transport problem and conditional history transport problem) are introduced. Each one reduces to the original problem of communicating an unstable process. In Section IV-B, it is shown that the anytime-reliable bit-transport problem reduces to the first problem (endpoint transport) in the pair. Finally, Section IV-C finishes the necessity argument by showing how traditional Shannon-reliable bit-transport reduces to the second problem and that the two of them can be put together. This reduces a pair of data-communication problems — anytime-reliable bit transport and Shannon-reliable bit-transport — to the original problem of communicating an unstable process to the desired fidelity.

The entire picture is illustrated in Fig. 6. Two data streams need to be embedded — a priority stream that requires anytime reliability and a remaining stream for which Shannon-reliability is good enough. The priority stream is used to generate the endpoints while the the history part is filled in with the appropriate conditional distribution. This simulated process is then run through the joint source-channel encoder $\mathcal{E}^s$ to generate channel inputs. The channel outputs are given to the joint source-channel decoder $\mathcal{D}^s$ which produces, after some delay ϕ, a fidelity d reconstruction of the simulated unstable process. By looking at the reconstructions corresponding to the endpoints, we are able to recover the priority data bits in an anytime reliable fashion. With these in hand, the remaining stream can also be extracted from the historical reconstructions.

A. Endpoints and history

This section parallels Section II, except from the other direction. Two analog problems are considered:

[18]For example a wireless channel subject to flat fading with the random fade fixed for all time. This channel has no Shannon capacity, but if it is good enough often enough it can be used to achieve good expected distortion.

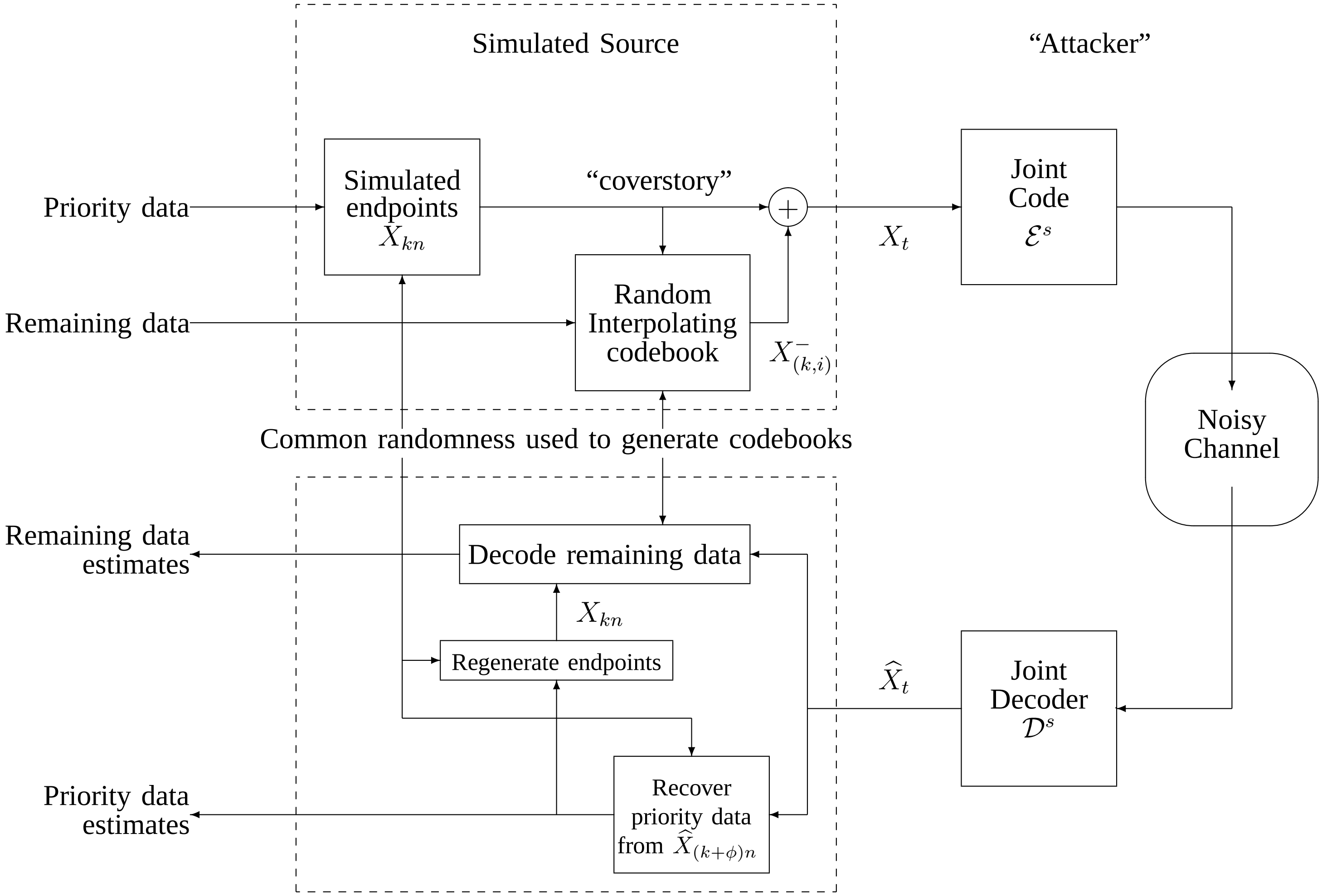

Fig. 6. Turning a joint-source-channel code into a two-stream code using information embedding. The good joint-source-channel code is like an attacker that will not impose too much distortion. Our goal is to create a source that carries our messages with two priorities. The priority data gets anytime reliability while the remaining data merely gets a low probability of error. The priority data is used to generate the endpoints of blocks of the unstable X. These endpoints are used to index into the random codebook for the remaining data. The output of the random codebook is an X^- realization that interpolates between the endpoints. It is combined with the endpoints to get a simulation for the unstable X process. The decoder recovers the priority bits from the reconstructed endpoints, and uses them to regenerate the original block endpoints. These are then used to decode the remaining data.

a) The endpoint problem: : Transportation of the process $\{X_{kn}\}$ where each sample arrives every n time steps and the samples are related to each other through (4) with $\tilde{W}_k$ being iid and having the same distribution as $A^{n-1}\sum_{i=0}^{n-1} A^{-i}W_i$.

This process must be transported so that $E[|X_{kn}-\widehat{X}_{kn}|^\eta] \leq K$ for some constant K. This is essentially a decimated version of the original problem.

b) The conditional history problem:: Given n, this is the problem of transporting an iid sequence of n-vectors $\vec{X}^-_k = (X^-_{k,1}, \ldots, X^-_{k,n-1})$ conditioned on iid Z_k that are known perfectly at the transmitter and receiver. The joint distribution of $\vec{X}^-, Z$ are given by:

$$\begin{aligned}
Z &= \sum_{t=0}^{n-1} A^{-t}W_t \\
X^-_{n-1} &= -A^{-1}W_{n-1} \\
X^-_t &= A^{-1}X^-_{t+1} - A^{-1}W_t
\end{aligned}$$

where the $\{W_t\}$ are iid. Expanding the recursion, we have $X_t^- = -\sum_{i=0}^{n-1-t} A^{-i-1} W_{t+i}$. The vectors $\vec{X}_k^-$ are made available to the transmitter every n time units along with their corresponding side-information Z_k. The goal is to transport these to the receiver so that $\frac{1}{n}\sum_{i=1}^{n-1} E[\rho(X_{k,i}^-, \widehat{X}_{k,i}^-)] \leq d$ for all k.

It is clear that the $\{Z_k\}$ in this second "conditional history" problem are essentially just scaled-down (by a factor of $A^{-(n-1)}$) versions of the $\{\tilde{W}_k\}$ from the first "endpoints" problem. The forward $\vec{X}_k = (X_{k,1}, \ldots, X_{k,n-1})$ can also be recovered using a simple translation of $\vec{X}_k^-$ by the vector $(Z_k, AZ_k, \ldots, A^{n-1}Z_k)$ since:

$$\begin{aligned} X_t &= \sum_{i=0}^{t-1} A^{t-i-1} W_i \\ &= \sum_{i=0}^{n-1} A^{t-i-1} W_i - \sum_{i=t}^{n-1} A^{t-i-1} W_i \\ &= A^{t-1} \sum_{i=0}^{n-1} A^{-i} W_i - \sum_{i=0}^{n-1-t} A^{-i-1} W_{t+i} \\ &= A^{t-1} Z + X_t^- \end{aligned}$$

It is obvious how to put these two problems together to construct an unstable X_t stream with a sample becoming available to the transmitter at every time unit: the endpoints problem provides the skeleton and the conditional history interpolates in between. To reduce the endpoints problem to the original unstable source transport problem, just use randomness at the transmitter to sample from the interpolating distribution and fill in the history.

To reduce the conditional history problem to the original unstable source transport problem, just use the iid Z_k to simulate the endpoints problem and use the interpolating $\vec{X}$ history to fill out $\{X_t\}$. Because the distortion measure is a difference distortion measure, the perfectly known endpoint process lets us translate everything so that the the same average distortion is attained.

B. Necessity of Anytime Reliability

We follow the spirit of information embedding[32] except that we have no a-priori covertext. Instead we use a simulated unstable process that uses common randomness and data bits assumed to be from iid coin tosses[19] to generate the endpoint process. This section also parallels the necessity story in [5], except that in this estimation context we have the additional complication of having a specified distribution for $\{W_t\}$, not just a bound on the allowed $|W_t|$.

The result is proved in stages. First, we assume that the density of W is a continuous uniform random variable plus something independent. After that, this assumption is relaxed to a Riemann-integrable density f_W.

1) Uniform driving noise:

Theorem 4.1: Anytime reliability is necessary for the endpoint stream:

If a joint source-channel encoder/decoder pair exists for the endpoint process given by (4) that achieves:

$$E[|X_{kn} - \widehat{X}_{kn}|^\eta] \leq d \tag{24}$$

for every position k, then for every rate $R < \log_2 A$, there exists an anytime code with common randomness for the channel that achieves anytime reliability of $\alpha = \eta \log_2 A$.

We further make the technical assumption that the original joint source-channel code is able to meet (24) for the process of (4) when driven by an iid noise distribution $W = G + U_\delta$ where G, U_δ are independent random variables with U_δ being a uniform random variable on the interval $[-\frac{\delta}{2}, +\frac{\delta}{2}]$ for some $\delta > 0$.

[19] If the data bits are not fair coin tosses to begin with, XOR them with a one-time pad using common randomness before embedding them.

Proof: Pick the initial condition X_0 using common randomness so it can be ignored in what follows. Since only the endpoints are going to matter for this theorem, pick some n large enough so that nR is an integer[20] with some target $R < \log_2 A$.

At the encoder, the goal is to simulate the endpoint process by simulating

$$\begin{aligned}\tilde{W}_k &= A^{n-1}\sum_{i=0}^{n-1} A^{-i}W_{k,i} \\ &= A^{n-1}W_{k,0} + A^{n-1}\sum_{i=0}^{n-1} A^{-i}W_{k,i} \\ &= A^{n-1}U_{\delta,k} + A^{n-1}(G_k + \sum_{i=0}^{n-1} A^{-i}W_{k,i}) \\ &= U_{A^{n-1}\delta,k} + A^{n-1}(G_k + \sum_{i=0}^{n-1} A^{-i}W_{k,i})\end{aligned}$$

The $A^{n-1}(G_k + \sum_{i=0}^{n-1} A^{-i}W_{k,i})$ term is simulated entirely using common randomness and is hence known to both the transmitter and receiver. The $U_{A^{n-1}\delta,k}$ term is a uniform random variable on $[-\frac{A^{n-1}\delta}{2}, +\frac{A^{n-1}\delta}{2}]$ and is simulated using a combination of common randomness and data bits. For the sake of exposition, assume that $\frac{A^{n-1}\delta}{2}$ is a power of two. Then we can write $U_{A^{n-1}\delta,k} = \frac{A^{n-1}\delta}{2}\sum_{\ell=1}^{\infty}(\frac{1}{2})^{\ell}S_{k,\ell}$ where the $S_{k,\ell}$ are iid random variables taking on values ± 1 each with probability $\frac{1}{2}$.

The idea is to embed the iid nR message bits into positions $\ell = 1, 2, \ldots, nR$ while letting the rest — a uniform random variable $U'_{\delta 2^{nR},k}$ representing the semi-infinite sequence of bits $(S_{k,nR+1}, S_{k,nR+2}, \ldots)$ — be chosen using common randomness. The result is:

$$\tilde{W}_k = A^{n-1}\frac{\delta}{2}M_k + A^{n-1}(U'_{\delta 2^{nR},k}G_k + \sum_{i=0}^{n-1} A^{-i}W_{k,i}) \tag{25}$$

where M_k is the nR bits of the message as represented by 2^{nR} equally likely points in the interval $[-1, +1]$ spaced apart by 2^{1-nR}, and the rest of the terms are chosen using common randomness known at both the transmitter and receiver side.

Since the simulated endpoints process is a linear function of the $\{\tilde{W}_k\}$, it suffices to just consider the $\{X'_{kn}\}$ process representing the response to the discrete messages $\{M_k\}$ alone. This has a zero initial condition and evolves like:

$$X'_{(k+1)n} = A^n X'_{kn} + \beta M_k \tag{26}$$

where $\beta = A^{n-1}\frac{\delta}{2}$. Expanding this recursion out as a sum:

$$X'_{(k+1)n} = (A^n)^k \beta \sum_{i=0}^{k} A^{-ni}M_{k-i} \tag{27}$$

which looks like a generalized binary expansion in base A^n and therefore implies that the X' process takes values on a growing Cantor set (illustrated in Fig. 7 for $nR = 1$)

The key property is that there are gaps in the Cantor set:

Property 4.1: If the rate $R < \log_2 A + \frac{\log_2(1-A^{-n})}{n}$ and the message-streams M and $\bar{M}$ first differ at position j (message $M_j \neq \bar{M}_j$), then at time $k > j$, the encoded X'_{kn} and $\bar{X}'_{kn}$ corresponding to the M_1^{k-1} and $\bar{M}_1^{k-1}$ respectively differ by at least:

$$|X'_{kn} - \bar{X}'_{kn}| \leq K A^{n(k-j)} \tag{28}$$

[20]Here, we ignore the irrational R case to avoid notational difficulties. [14] and [5] show how the argument can extend naturally to the irrational case.

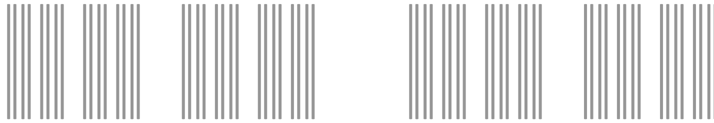

Fig. 7. The priority data bits are used to refine a point on a Cantor set. The natural tree structure of the Cantor set construction allows us to encode bits sequentially. The Cantor set also has finite gaps between all points corresponding to bit sequences that first differ in a particular bit position. These gaps allow us to reliably extract bit values from noisy observations of the Cantor set point regardless of which point it is.

for some constant $K > 0$ that does not depend on the values of the data bits, k, or j.

This is true since:

$$\begin{aligned} |X'_{kn} - \bar{X}'_{kn}| &\geq A^{n(k-j)}\beta(|M_j - \bar{M}_j| - \left|\sum_{i=j+1}^{\infty} A^{-n(i-j)}(M_i - \bar{M}_i)\right|) \\ &\geq A^{n(k-j)}\beta(|M_j - \bar{M}_j| - 2A^{-n}\sum_{i=0}^{\infty} A^{-ni}) \\ &\geq A^{n(k-j)}\beta(2^{1-nR} - 2\frac{A^{-n}}{1-A^{-n}}) \\ &= A^{n(k-j)}2\beta(2^{-nR} - \frac{1}{A^n - 1}) \end{aligned}$$

Which is positive as long as $2^{-nR} > \frac{1}{A^n-1}$ or $nR < \log_2 A^n - 1$. We can thus use $K = 2\beta(2^{-nR} - \frac{1}{A^n-1}) = \frac{\delta}{A}(2^{n(\log_2 A - R)} - \frac{A^n}{A^n-1})$

In coding theory terms, Property 4.1 can be interpreted as an infinite Euclidean free-distance for the code with the added information that the distance increases exponentially as $A^{n(k-j)}$. An error can only happen if the received "codeword" is more than half the minimum distance away.

At the decoder, the common randomness means that the estimation error $X_{kn} - \widehat{X}_{kn}$ is the error in estimating X'_{kn}. By applying Markov's inequality to the error using (24), we immediately get a bound on the probability of an error on the prefix M_0^i for $i < k$:

$$\begin{aligned} \mathcal{P}(\widehat{M}_1^i(kn) \neq M_1^i) &\leq \mathcal{P}(|\widehat{X}'_{kn} - X'_{kn}| > \frac{K}{2}A^{n(k-i)}) \\ &= \mathcal{P}(|\widehat{X}_{kn} - X_{kn}| > \frac{K}{2}A^{n(k-i)}) \\ &= \mathcal{P}(|\widehat{X}_{kn} - X_{kn}|^\eta > (\frac{K}{2})^\eta (A^{n(k-i)})^\eta) \\ &\leq d(\frac{K}{2})^{-\eta}(A^{n(k-i)})^\eta \\ &= K' 2^{-(\eta \log_2 A) n(k-i)} \end{aligned}$$

But $n(k-i)$ is the delay that is experienced at the message level[21] and so the desired anytime reliability is obtained. To approach $R = \log_2 A$ as closely as desired, n can be increased as needed until $R < \log_2 A + \frac{\log_2(1-A^{-n})}{n}$ □

2) General driving noise: Theorem 4.1 can have the technical smoothness condition weakened to simply requiring a Riemann-integrable density for the white W driving process.

Theorem 4.2: Anytime reliability is necessary for the endpoint stream:

[21]If bits have to be buffered-up to form messages, then the delay at the bit level includes another constant n. This only increases the constant K' further but does not change the exponent with large delays.

If the W have a Riemann-integrable density f_W and a joint source-channel encoder/decoder pair exists for the endpoint process given by (4) that achieves:

$$E[|X_{kn} - \widehat{X}_{kn}|^\eta] \le d$$

for every position k, then for every rate $R < \log_2 A$, there exists an anytime code with common randomness for the channel that achieves any anytime reliability $\alpha < \eta \log_2 A$.

Furthermore, the receiver can actually learn the endpoint process with zero distortion with a probability tending to 1 as the delay increases beyond the delay induced by the code for the endpoint process.

Proof: Since the density is Riemann-integrable, f_W can be expressed as a non-negative piecewise constant function f'_W that only changes at integer multiples of δ plus a non-negative function f''_W representing the "error" in Riemann-integration from below. By choosing δ small enough, the total mass in f''_W can be made as small as we want since the Riemann sums must converge.

Choose δ such that the total mass in f''_W is $\gamma \le A^{-2\eta n}$. W can be simulated in the following way:

1) Flip an independent biased coin with probability of heads γ.
2) If heads, independently draw the value of W from the density $\frac{1}{\gamma} f''_W$
3) If tails, independently draw the value of W from the piecewise constant density $\frac{1}{1-\gamma} f'_W$. This can be done by using a discrete random variable G plus an independent uniform random variable U_δ.

When simulating $W_{k,0}$ in the endpoint process, use common randomness for steps 1 and 2, while following the procedure from the proof of Theorem 4.1 for step 3. We can interpret a "heads" in step 1 as an "erasure" with probability γ since no message can be encoded in that time period. Since the outcome of these coin tosses come from common randomness, the position of these erasures are known to both the transmitter and the receiver. In this way, it behaves like a packet erasure channel with a very low probability of erasure. This problem is studied in Section VI of [7], and the delay-optimal coding strategy relative to the erasure channel is to place incoming packets into a FIFO queue awaiting a non-erased opportunity for transmission. The following coarse lemma summarizes the results we need from [7] and corresponds to equation (31), Corollary 6.1, and Fig. 13 in [7].

Lemma 4.1: Low-rate generalized erasure channels: Suppose packets arrive deterministically at a rate of R per unit time and enter a FIFO queue drained at constant rate 1.

- If each packet has a size distribution that is bounded below a geometric$(1-\gamma)$ (i.e. $P(\text{Size} > s) \le \gamma^s$ for all non-negative integers s), then the random delay ϕ experienced by any individual packet in the queue satisfies $P(\text{Delay} > s) \le K 2^{-\alpha s}$ for all non-negative s and some constant K that does not depend on s. Furthermore, if $R < \frac{1}{1+2r}$ for some $r > 0$, then $\alpha \ge -\log_2 \gamma - \gamma^r$.
- If the rate $R = \frac{1}{n}$ and each packet has a size distribution that is bounded by: $P(\text{Size} > n(1-\epsilon)+s) \le \gamma^s$ for all non-negative integers s, then the excess[22] delay ϕ experienced by any individual packet has the same tail distribution as that for $R' = \frac{1}{n\epsilon}$ and packets with geometric$(1-\gamma)$ size.

In this problem, suppose the message bits are arriving deterministically at bit-rate $R < \log_2 A$ per unit time to the transmitter. Pick $r > 0$ small enough so that $R' = (1+3r)R < \log_2 A$. Group message bits into packets of size nR'. These packets arrive deterministically at rate $\frac{1}{1+3r} < \frac{1}{1+2r}$ packets per n time units. Thus, Lemma 4.1 applies and the delay (in n units) experienced by a packet in the queue has an exponential tail with an exponent of least

$$\begin{aligned} -\log_2 \gamma - \gamma^r &\ge -\log_2 A^{-2\eta n} - A^{-2\eta n r} \\ &= n 2\eta \log_2 A - A^{-2\eta n r} \end{aligned}$$

per n time steps or $2\eta \log_2 A - \frac{A^{-2\eta n r}}{n}$ per unit time step. When n is large, this exponent is much faster than the delay exponent of $\eta \log_2 A$ obtained in the proof of Theorem 4.1. Thus the dominant delay-exponent remains $\eta \log_2 A$ as desired.

[22]Beyond the minimum $n(1-\epsilon)$ required even when the queue is empty when it arrives.

Finally, the simulated endpoint process depends only on common randomness and the message packets. Since the common randomness is known perfectly at the receiver by assumption and the message packets are known with a probability that tends to 1 with delay, the endpoint process is also known with zero distortion with a probability tending to 1 as the delay increases and the receiver. □

The significance of Theorem 4.2 is that exponentially unstable processes fundamentally contain a rate $\log_2 A$ bits of information per time that is nonclassical in nature. This information requires anytime reliability for transport by a communication system where the sense of reliability depends not just on the process, but also on the stringency of the fidelity criterion. For example, requiring a finite fourth moment of error $\eta = 4$ results in twice the required reliability as compared to requiring only a finite second moment $\eta = 2$. Interestingly, only the finiteness of the error moment matters for the exponent, not the actual distortion targeted.

C. *Embedding classical bits*

Our goal is to show that any channel or communication system used to transport unstable Markov process to average distortion d must support a classically-reliable data stream of rate $\approx R(d) - \log_2 A$ in addition to the essential rate $\approx \log_2 A$ stream with anytime reliability in the previous section.

Classical rate-distortion points out that the mutual information across the communication system must be at least $R(d)$ on average and so from that perspective, the room is there for the new stream. However, as [33] points out, having enough mutual information is not enough to guarantee a reliable-transport capacity since the virtual channel facing the residual data stream in Fig. 6 is not stationary and ergodic. Consequently, a low-enough expected distortion is not enough.

To remedy this, an additional condition must be placed on the joint source/channel code for unstable processes. We require that the probability of excess average distortion over any long enough segment tends to zero.

Theorem 4.3: Suppose there exists a family of black-box systems (viewed as joint source/channel codes $(\mathcal{E}^s, \mathcal{D}^s)$) for the unstable process given by (1) so that each member of the family satisfies all of the assumptions of Theorem 4.2 including (24) and the family (indexed by window size n) also satisfies for all τ:

$$\lim_{n\to\infty} \sup_{\tau\geq 0} \mathcal{P}(\frac{1}{n}\sum_{i=\tau}^{\tau+n-1} |\widehat{X}_i - X_i|^\eta > d) = 0 \tag{29}$$

Then by appropriately embedding data into a simulated $\{X_t\}$ process, in addition to carrying a rate $R_1 < \log_2 A$ priority stream with anytime reliability, the communication system can also be made to carry a second stream of data at any rate $R_2 < R^X_\infty(d) - \log_2 A$ so that the probability of bit error is as low as desired.

Proof: The overall construction is described in Fig. 6. ϵ is chosen small enough and n is chosen to be large enough so that the R_1 stream can be successfully embedded in the endpoint process by Theorem 4.2, as well as being large enough so that $R_2 < R_n^{X|X_n}(d+\epsilon)$ the conditional rate-distortion function for the history given the endpoint.

By choosing an appropriate additional delay, Theorem 4.2 assures us that the receiver will know all the past simulated endpoints correctly with an arbitrarily small probability of error. As described in Section IV-A, this means we now have a family of systems (indexed by m) that solve the conditional history problem with the further guarantee that:

$$\lim_{m\to\infty} \sup_{k\geq 0} \mathcal{P}(\frac{1}{m}\sum_{k=\tau}^{\tau+m-1} \frac{1}{n}\sum_{i=1}^{n-1} |X^-_{(k,i)} - \widehat{X}^-_{(k,i)}|^\eta > d) = 0 \tag{30}$$

It tells us that by picking m large enough, the probability of having excess distortion can be made as small as desired.

The $\{Z_k\}$ are interpreted as the "coverstory" that must be respected when embedding data into the $\{\vec{X}_k^-\}$ process. Theorem 3 from [18] (full proofs in [19]) applies and tells us that a length $m' > m$ random code with $\vec{X}_k^-$ drawn independently of each other, but conditional on the iid Z_k, can be used to embed information at any rate $nR_2 < nR_n^{X^-|Z}(d+\epsilon) = nR_n^{X|X_n}(d+\epsilon)$ per vector symbol with arbitrarily low probability of error. □

The "weak law of large numbers"-like condition (29), or something like it, is required for the theorem to hold since there are joint source-channel codes for which mutual information can not be turned into the reliable transport of bits at arbitrarily low probabilities of error. Consider the following example. Suppose we had two different joint source-channel codes available: one of which had a target distortion of d_1 and the other of which had a target distortion of $d_2 = 10d_1$. The actual joint code, which is presumed to have access to common randomness, could decide with probability $\frac{1}{1000}$ to use the second code rather than the first. In such a case, the ensemble average mutual information is close to $R(d_1) - \log_2 A$ bits, but with non-vanishing probability $\frac{1}{1000}$ we might not be able sustain such a rate over the virtual channel.

We conjecture that for DMCs, if any joint source-channel code exists that hits the target distortion, then one should also exist that meets (29) and it should be possible to simultaneously transport two streams of data reliably with sufficient anytime reliability on the first stream and enough residual rate on the second.

V. Unstable Gaussian Markov processes with squared-error distortion

The main challenge in the Gaussian case is that the disturbance does not have bounded support. Before extending the theorems to cover the Gaussian case, it is interesting to plot the rate-distortion functions themselves.

A. Comparison to the sequential distortion-rate function

In the case of Gauss-Markov processes with squared-error distortion, Hashimoto and Arimoto in [10] give us an explicit way of calculating $R(d)$. Tatikonda in [22], [34] gives an explicit way of doing a similar calculation when we force the reconstruction $\widehat{X}_t$ to be causal and depend only on X_j observations for $j \leq t$.

Assuming unit variance for the driving noise W, Hashimoto's formula is parametric in terms of the water-filling parameter κ and for the Gauss-Markov case considered here simplifies to:

$$D(\kappa) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \min\left[\kappa, \frac{1}{1 - 2A\cos(\omega) + A^2}\right] d\omega \quad (31)$$

$$R(\kappa) = \log_2 A + \frac{1}{2\pi} \int_{-\pi}^{\pi} \max\left[0, \frac{1}{2}\log_2 \frac{1}{\kappa(1 - 2A\cos(\omega) + A^2)}\right] d\omega \quad (32)$$

The corresponding stable case of the backwards process has a water-filling solution that is identical, except without the $\log_2 A$ term in the $R(\kappa)$. In the Gaussian case, (18) holds with equality.

Under the same assumptions, Tatikonda's formula for the case with causal reconstructions becomes:

$$R_{\text{seq}}(d) = \frac{1}{2}\log_2(A^2 + \frac{1}{d}) \quad (33)$$

Fig. 8 shows the distortion-rate frontier for the forwards process and backwards process in the Gaussian case. It is easy to see that the forwards and backwards process curves are translations of each other. In addition, the sequential rate-distortion curve for the forward process is qualitatively distinct in behavior. $D_{\text{seq}}(R)$ goes to infinity as $R \downarrow \log_2 A$ while $D(R)$ stays finite. There still remains a bit of a mystery regarding the true nature of this gap. The plot suggests that it takes some time for the randomness entering the unstable process through W to sort itself into the two categories of fundamental accumulation and transient history. In particular, it is open whether a similar information-embedding theorem can be given that gives an operational meaning to the gap between $R_{seq}(d)$ and $R(d)$.

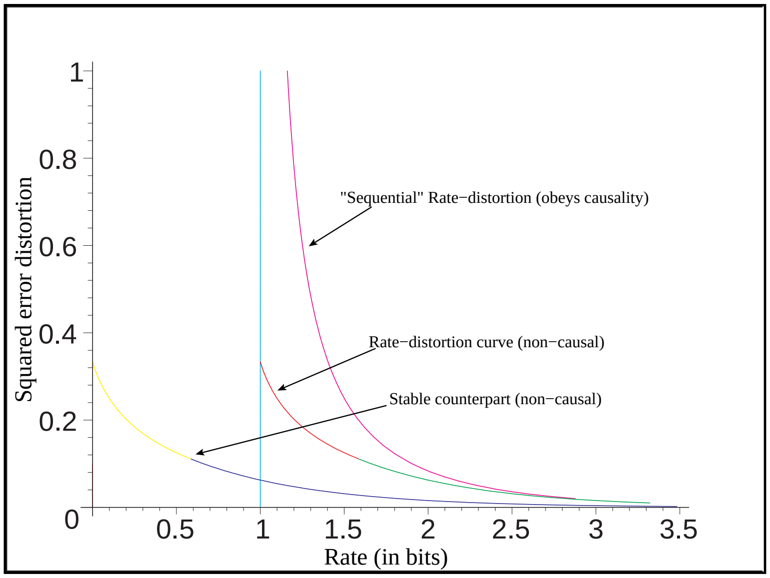

Fig. 8. The distortion-rate curves for an unstable Gauss-Markov process with $A = 2$ and its stable backwards-version. The stable and unstable $D(R)$ curves are related by a simple translation by 1 bit per symbol.

B. Source-coding for Gaussian Processes

The counterpart to Theorem 2.1 is:

Corollary 5.1: Given an unstable ($A > 1$) scalar Markov process as given by (1) driven by independent Gaussian noise $\{W_t\}_{t\geq 0}$ with zero mean and variance σ^2, it is possible to encode the process to average fidelity arbitrarily close to d using two fixed-rate bitstreams. By choosing a sufficiently large end-to-end delay, the first stream can be made to have rate R_1 arbitrarily close to $\log_2 A$ while the second can have rate R_2 arbitrarily close to $R^X_\infty(d) - \log_2 A$.

Proof: The strategy is essentially as before. One simplification is that we can make full use of the results of Section II-F since in the Gaussian case, the direct computations in the previous section reveal that $R^{X-}_\infty(d) = R^X_\infty(d) - \log_2 A$. There is no rate loss in encoding the historical segments on a block-by-block basis rather than using superblocks and conditional encodings. The main issue is dealing with the unbounded support when encoding the checkpoints.

(a) Look at time in blocks of size n and encode the values of checkpoints X_{kn} recursively to very high precision using a prefix-free variable-length code with rate $n(\log_2 A + \epsilon_1) + L_k$ bits per value, where the L_k are iid random variables with nice properties.

(b) Smooth out the variable-length code by running it through a FIFO queue drained at constant rate $R_1 = \log_2 A + \epsilon_1 + \epsilon_q$. Make sure that the delay exponent in the queue is high enough.

(c) Use the exact value for the ending checkpoint $X_{(k+1)n}$ to transform the segment immediately before it so that it looks exactly like a stable backwards Gaussian process of length n with initial condition 0. Encode each block of the backwards history process to average-fidelity d using a fixed-rate rate-distortion code for the backwards process that operates at rate $R^{X-}_\infty(d) + \epsilon_s$.

(d) At the decoder, wait ϕ time units and attempt to decode the checkpoints to high fidelity. If the FIFO queue is running too far behind, then extrapolate a reconstruction based on the last checkpoint we were able to fully decode.

(e) Decode the history process to average-fidelity d and combine it with the checkpoints to get the reconstruction.

Offset	Codeword
0	100
+1	1110
-1	1100
+2	11110
-2	11010
+3	111110
-3	110110
+4	1111110
-4	1101110
⋮	⋮

Fig. 9. Unary encoding of integer offsets to deal with the unbounded support. The first bit denotes start while the nest two bits reflect the sign. The length of the rest reflects the magnitude of the offset with a zero termination. The encoding is prefix-free and hence uniquely decodable. The length of the encoding of integer S is bounded by $3 + |S|$

a) Encoding the checkpoints: (4) remains valid, but the term $\tilde{W}_k = A^{n-1}\sum_{i=0}^{n-1} A^{-i}W_{kn+i}$ is not bounded since the W_i are iid Gaussians. The $\tilde{W}_k$ are instead Gaussian with variance:

$$\begin{aligned}\tilde{\sigma}^2 &= A^{2(n-1)}\sum_{i=0}^{n-1} A^{-2i}\sigma^2 \\ &\leq A^{2(n-1)}\sigma^2\sum_{i=0}^{\infty} A^{-2i} \\ &= A^{2n}\frac{\sigma^2}{A^2-1}\end{aligned}$$

Its standard deviation $\tilde{\sigma}$ is therefore $A^n\frac{\sigma}{\sqrt{A^2-1}}$. Pick $l = 2^{\frac{\epsilon_1}{3}n}$ and essentially pretend that this random variable $\tilde{W}_k$ has bounded support of $l\tilde{\sigma}$ during the encoding process. By comparing (5) to the above, the effective Ω is simply $l\sigma\frac{2(A-1)}{\sqrt{A^2-1}} = 2^{\frac{\epsilon_1}{3}n}\sigma\sqrt{\frac{A-1}{A+1}}$. Define $\tilde{\Omega} = \sigma\sqrt{\frac{A-1}{A+1}}$ so that the effective $\Omega = 2^{\frac{\epsilon_1}{3}n}\tilde{\Omega}$.

Encode the checkpoint increments recursively as before, only add an additional variable length code to encode the value of $\lfloor\frac{\tilde{W}}{l\tilde{\sigma}} + \frac{1}{2}\rfloor$ while treating the remainder using the fixed-rate code as before. The variable length code is a unary encoding that counts how many $l\sigma$ away from the center the $\tilde{W}_k$ actually is. (Fig. 9 illustrates the unary code.) Let L_k be the length of the k-th unary codeword, then this is bounded above by:

$$P(L_k \geq 3 + j) \;=\; P(|\tilde{W}| > jl\tilde{\sigma})$$

Let N be a standard Gaussian random variable and rewrite this as:

$$P(L_k \geq 3 + j) = P(|N| > j2^{\frac{\epsilon_1}{3}n}) \tag{34}$$

and so L_k is very likely indeed to be small and certainly has a finite expectation.

The fixed-rate part of the checkpoint encoding has a rate that is the same as that given by (6), except that Ω is now mildly a function of n. Plugging in $2^{\frac{\epsilon_1}{3}n}\tilde{\Omega}$ for Ω in (6) gives:

$$\begin{aligned}
R_{1,f} &\geq \max\left(\log_2 A + \frac{\log_2(1+\frac{\Omega}{\Delta(A-1)}) + \log_2(2+\frac{\Omega}{\Delta})}{n}, \frac{\log_2\lceil\frac{\Omega_0}{\Delta}\rceil}{n}\right) \\
&= \max\left(\log_2 A + \frac{\log_2(1+\frac{2^{\frac{\epsilon_1}{3}n}\tilde{\Omega}}{\Delta(A-1)}) + \log_2(2+\frac{2^{\frac{\epsilon_1}{3}n}\tilde{\Omega}}{\Delta})}{n}, \frac{\log_2\lceil\frac{\Omega_0}{\Delta}\rceil}{n}\right) \\
&= \max\left(\log_2 A + \frac{2}{3}\epsilon_1 + \frac{\log_2(2^{-\frac{\epsilon_1}{3}n}+\frac{\tilde{\Omega}}{\Delta(A-1)}) + \log_2(2^{1-\frac{\epsilon_1}{3}n}+\frac{\tilde{\Omega}}{\Delta})}{n}, \frac{\log_2\lceil\frac{\Omega_0}{\Delta}\rceil}{n}\right)
\end{aligned}$$

Essentially, the required rate $R_{1,f}$ for the fixed-rate part has only increased by a small constant $\frac{2}{3}\epsilon_1$. Holding Δ fixed and assuming n is large enough, we can see that

$$R_{1,f} = \log_2 A + \epsilon_1 \tag{35}$$

is sufficient.

b) Smoothing out the flow: The code so far is variable-rate and to turn this into a fixed-rate $R_1 = \log_2 A + \epsilon_1 + \epsilon_q$ bitstream, it is smoothed by going through a FIFO queue. First, encode the offset using the variable-length code and then recursively encode the increment as was done in the finite support case. All such codes will begin with a 1 and thus we can use zeros to pad the end of a codeword whenever the FIFO is empty. After all the average input rate to the FIFO is smaller than the output rate and hence it will be empty infinitely often.

c) Getting history and encoding it: Section II-F explains why such a transformation is possible by subtracting off a scaled version of the endpoint. The result is a stable Gaussian process and so [9] reveals that it can be encoded arbitrarily close to its rate-distortion bound $R_\infty^{X^-}(d)$ if n is large enough. For the unstable Gaussian Markov process, $R_\infty^{X^-}(d) = R_\infty^X(d) - \log_2 A$ as we saw earlier.

d) Decoding the checkpoints: At the decoder, we can wait long enough so that the checkpoint we are interested in is very likely to have made it through the FIFO queue by now. The ideas here are similar to [7], [35] in that we are using a FIFO queue to smooth out the rate variation and are interested in its large deviations performance. There is $n\epsilon_q$ slack that has to accommodate L_k bits. Because n can be made large, the error exponent with delay here is as large as we want it to be.

More precisely, a packet of size $n(\epsilon_1 + \log_2 A) + L_k$ bits arrives every n time units where the L_k are iid. This is drained at rate $R_1 = \epsilon_q + \epsilon_1 + \log_2 A$. An alternative view is therefore that a point packet arrives deterministically every n time units and it has a random service time T_k given by $n\frac{\epsilon_1+\log_2 A}{\epsilon_q+\epsilon_1+\log_2 A} + \frac{L_k}{\epsilon_q+\epsilon_1+\log_2 A}$. Define $(1-\epsilon_q') = \frac{\epsilon_1+\log_2 A}{\epsilon_q+\epsilon_1+\log_2 A}$. Then the random service time $T_k = (1-\epsilon_q')n + \frac{L_k}{\epsilon_q+\epsilon_1+\log_2 A}$ when measured in time units or $T_k^b = (1-\epsilon_q')nR_1 + L_k$ when measured in bit-units.

This can be analyzed using large-deviations techniques or by applying standard results in queuing. The important thing is a bound on the length L_k which is provided by (34). Since the tail probability of a standard normal dies at least as fast as some exponential[23], it is clear that

$$\begin{aligned}
P(L_k \geq 3 + j) &= P(|N| > j2^{\frac{\epsilon_1}{3}n}) \\
&\leq \exp(-(\alpha' 2^{\frac{\epsilon_1}{3}n})j)
\end{aligned}$$

Since an exponential eventually dominates all constants, we know that for any $\beta > 0$, there exists a sufficiently large n so that:

$$P(L_k - 2 > j) \leq 2^{-\beta j} \tag{36}$$

[23]This is done to illustrate that while this proof is written for the Gaussian case, the arguments here readily generalize to any driving distribution W that has at least an exponential tail probability. To accommodate W with power-law tail distributions would require the use of logarithmic encodings as described in [36], [37]. This does not work for our case because the unary nature to the encoding is important when we consider transporting such bitstreams across a noisy channel.

Thus, the delay (in bits) experienced by a block in the queue will behave no worse than that of point messages arriving every nR_1 bits where each requires at least $nR_1(1-\epsilon'_q)+2 = nR_1(1-\epsilon''_q)$ bits plus an iid geometric$(1-p)$ number of bits with $p = 2^{-\beta}$.

Once again, Lemma 4.1 applies to this queuing problem and the second part of that lemma tells us that the delay performance is exactly the same as that of a system with point messages arriving every $n\epsilon''_q$ bits requiring only an iid geometric number of bits. Since $\frac{1}{n\epsilon''_q}$ is small, the first part of Lemma 4.1 applies. If we pick $r = \frac{n\epsilon''_q}{3} - 1$, then the bit-delay exponent[24] α_b is at least

$$\begin{aligned} \alpha_b &\geq -\log_2 2^{-\beta} - 2^{-\beta r} \\ &= \beta - 2^{-\beta(\frac{n\epsilon''_q}{3}-1)} \end{aligned}$$

which is at least $\beta - 1$ when $n\epsilon''_q \geq 3$. Converting between bit-delay and time-delay is just a factor of $\log_2 A$ and so the time-delay exponent is at least $\frac{\beta-1}{\log_2 A}$. But β can be made as large as we want by choosing n large enough.

e) Getting the final reconstruction: The history process can be added to the recovered checkpoint. This differs from the original process by only the error in the history plus the impact of the error in the checkpoint. The checkpoint reconstruction-error's impact dies exponentially since the history process is stable. So the target distortion is achieved if the checkpoint has completely arrived by time the reconstruction is attempted. By choosing a large enough end-to-end delay ϕ, the probability of this can be made as high as we like.

However, the goal of our source-code is not just to meet the target distortion level d with high probability, it is also to hit the target in expectation. Thus, we must bound the impact of not having the checkpoint available in time. When this happens, the un-interpretable history information is ignored and the most recent checkpoint is simply extrapolated forward to the current time. The expected squared errors grow as $A^{2\psi}$ where ψ is the delay in time-units. The arguments here exactly parallel those of Theorem 3.6, where the FIFO queue is acting like an anytime code. Since the delay-exponent of the queue is as large as we want, it can be made larger than $2\log_2 A$. Thus, the expected distortion coming from such "overflow" situations is as small as desired. This completes the proof of Corollary 5.1. $\square$

C. Channel sufficiency for transporting Gaussian Processes

With a noisy channel, the story in the Gaussian case is essentially unchanged since the historical information is as classical as ever. The only issue is with the checkpoint stream. An error in a bit ψ steps ago can do more than propagate through the usual pathway. It could also damage the bits corresponding to the variable-length offset. But because of the unary encoding[25] and the $2^{\frac{\epsilon_1}{3}n}$ expansion in the effective Ω, an uncorrected bit-stream error ψ time-steps ago can only impact the current reconstruction by an $O(\psi 2^{\frac{\epsilon_1}{3}n} A^{\psi})$ change in its value. The key is to understand that the delay ψ is much larger than the block-length n and so the polynomial term in front is insignificant relative to the exponential in ψ and so the story is unchanged. Thus we have:

Corollary 5.2: Anytime reliability plus classical reliability is sufficient to transport Gaussian processes: Suppose that a communication system provides us with the ability to carry two data streams. One at rate $R_1 > \log_2 A$ with uniform anytime reliability $\alpha > 2\log_2 A$, and another at rate $R_2 > R^X_\infty(d) - \log_2 A$ with classical Shannon reliability where $R^X_\infty(d)$ is the rate-distortion function for an unstable Gaussian Markov process with unstable gain $|A| \geq 1$. Then it is possible to successfully transport the two-stream code of Corollary 5.1 using this communication system by picking sufficient end-to-end delay ϕ. The mean squared error of the resulting system will be as close to d as desired.

[24]The bit-delay exponent α_b bounds the probability that immediately before the block in question entered the queue, that there were already Q bits awaiting transmission. Precisely, $P(Q > q) \leq K 2^{-\alpha_b q}$ where K is some constant that does not depend on q. Time-delay and bit-delay are in one-to-one correspondence since the queue is FIFO and is drained at a constant rate.

[25]A logarithmic sized encoding would be more sensitive to errors.

D. Necessity for Gaussian Processes

The Gaussian disturbance W already has a Riemann-integrable density and so Theorem 4.3 already applies.

VI. Extensions to the vector case

With the scalar case explored, it is natural to consider what happens for general finite-dimensional[26] linear models where A is a matrix and X is a vector. Though the details are left to the reader, the story is sketched here. No fundamentally new phenomena arise in the vector case, except that different anytime reliabilities can be required on different streams arising from the same source as is seen in the control context in [5].

The source-coding results here naturally extend to the fully observed vector case with generic driving noise distributions. Instead of two data streams, there is one special stream for each unstable eigenvalue of A and a singe final stream capturing the residual information across all dimensions. All the sufficiency results also generalize in a straightforward manner — each of the unstable streams requires a corresponding anytime reliability depending on the distortion function's η and the magnitude of the eigenvalue. The multiple priority-stream necessity results also follow generically.[27] This is a straightforward application of a system diagonalization[28] argument followed by an eigenvalue by eigenvalue analysis. The necessity result for the residual rate follows the same proof as here based on inverse-conditional rate-distortion with the endpoints in all dimensions used as side-information.

The case of partially observed vector Markov processes where the observations $C_y\vec{X}$ are linear in the system state requires one more trick. We need to invoke the observability[29] of the system state through the observations. Instead of a single checkpoint pair, use an appropriate number[30] of consecutive values for the observation and encode them together to high fidelity Δ for the sufficiency story. This can be done by transforming coordinates linearly so that the system is diagonal, though driven by correlated noise, from checkpoint-block to the next checkpoint-block. Each unstable eigenvalue will contribute its own $\log_2 \lambda_i$ term to the first stream rate and will require the appropriate anytime reliability. The overhead continues to be sublinear in n and the residual information continues to be classical in nature by the same arguments given here. The partially observed necessity story is essentially unchanged on the information embedding side, except that every long block should be followed by a miniblock of the appropriate length[31] during which no data is embedded and only common-randomness is used to generate the driving noise. This will allow the decoder to easily use the observability to get noisy access to the unstable state itself.

In [6], these techniques are applied in the context of control rather than estimation. The interested reader is referred there for the details. Some simplifications to the general story might be possible in the case of SISO autoregressive processes, but we have not explored them in detail.

VII. Conclusions

We have characterized the nature of information in an unstable Markov process. On the source coding side, this was done by giving an appropriate fixed-rate coding Theorem 2.1. This theorem's code construction naturally produces two streams — one that captures the essential unstable nature of the process

[26] In the Gaussian case, these will correspond to cases with rational power-spectral densities.

[27] The required condition is that the the driving noise distribution W should not have support isolated to an invariant subspace of A. If that were to happen, there would be modes of the process that are never excited.

[28] The case of non-diagonal Jordan blocks is only a challenge for the necessity part regarding anytime reliability. It is covered in [6] in the control context. The same argument holds here with a Riemann-integrable joint-density assumption on the driving noise.

[29] The linear observation should not be restricted to a single invariant subspace. If it were, we could drop the other subspaces from the model as irrelevant to the observed process under consideration.

[30] The appropriate number is twice the number of observations required before all of the unstable subspaces show up in the observation. This number is bounded above by twice the dimensionality of the vector state space. The factor of two is to allow each block to have its own beginning and end.

[31] Again, the dimensionality of the underlying state space suffices.

and requires a rate of at least $\log_2 A$, and another that captures the essentially classical nature of the information left over. The quantitative distortion is dominated by the encoding of the second stream, while the first stream serves to ensure its finiteness as time goes on. The essentially stable nature of the second stream's information was then shown by Theorem 2.2 which relates the forward $D(R)$ curve to the "backwards" one corresponding to a stable process.

At the intersection of source and channel coding, we reviewed the notion of anytime reliability and Theorem 3.5 showed that it is nonzero for DMCs at rates below capacity. Theorems 3.6 and 4.2 then showed that the first stream requires a high-enough anytime reliability for transport over a communication system rather than merely enough rate. In contrast, Theorems 3.7 and 4.3 showed that the second stream requires only sufficient rate. Together, all these results establish the relevant separation principle for such unstable processes.

This work brings the exponentially unstable processes firmly into the fold of information theory. More fundamentally, it shows that reliability functions are not a matter purely internal to channel coding. In the case of unstable processes, the demand for appropriate reliability arises at the source-channel interface. Thus unstable processes have the potential to be useful models while taking an information-theoretic look at QoS issues in communication systems. The success of the "reductions and equivalences" paradigm of [5], [19] here suggests that this approach might also be useful in understanding other situations in which classical approaches to separation theorems break down.

ACKNOWLEDGMENTS

The authors would like to give special thanks to Mukul Agarwal for the discussions regarding this paper and the resulting contributions in [19] where one of the key results required is proved. We also thank Nigel Newton, Nicola Elia, and Sekhar Tatikonda for several constructive discussions about this general subject over a long period of time which have influenced this work in important ways.

REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, July/Oct. 1948.
[2] ——, "Coding theorems for a discrete source with a fidelity criterion," *IRE National Convention Record*, vol. 7, no. 4, pp. 142–163, 1959.
[3] S. Vembu, S. Verdu, and Y. Steinberg, "The source-channel separation theorem revisited," *IEEE Trans. Inform. Theory*, vol. 41, no. 1, pp. 44–54, Jan. 1995.
[4] A. Ephremides and B. Hajek, "Information theory and communication networks: An unconsummated union," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2416–2434, Oct. 1998.
[5] A. Sahai and S. K. Mitter, "The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link. part I: scalar systems," *IEEE Trans. Inform. Theory*, Aug. 2006.
[6] ——, "The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link. part II: vector systems," *IEEE Trans. Inform. Theory*, submitted for publication. [Online]. Available: http://www.eecs.berkeley.edu/~sahai/Papers/control-part-II.pdf
[7] A. Sahai, "Why block length and delay are not the same thing," *IEEE Trans. Inform. Theory*, submitted. [Online]. Available: http://www.eecs.berkeley.edu/~sahai/Papers/FocusingBound.pdf
[8] R. G. Gallager, *Information Theory and Reliable Communication*. New York, NY: John Wiley, 1971.
[9] R. Gray, "Information rates of autoregressive processes," *IEEE Trans. Inform. Theory*, vol. 16, no. 4, pp. 412–421, July 1970.
[10] T. Hashimoto and S. Arimoto, "On the rate-distortion function for the nonstationary autoregressive process," *IEEE Trans. Inform. Theory*, vol. 26, no. 4, pp. 478–480, Apr. 1980.
[11] T. Berger, "Information rates of Wiener processes," *IEEE Trans. Inform. Theory*, vol. 16, no. 2, pp. 134–139, Mar. 1970.
[12] ——, *Rate Distortion Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
[13] A. Sahai, "A variable rate source-coding theorem for unstable scalar markov processes," in *Proceedings of the 2001 IEEE Symposium on Information Theory*, Washington, DC, June 2001.
[14] ——, "Any-time information theory," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2001.
[15] S. K. Mitter and N. J. Newton, "Information flow and entropy production in the Kalman-Bucy filter," *Journal of Statistical Physics*, vol. 118, no. 1, pp. 145–175, Jan. 2005.
[16] A. Sahai and S. K. Mitter, "A fundamental need for differentiated 'quality of service' over communication links: An information theoretic approach," in *Proceedings of the Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2000.
[17] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Reading, MA: Addison-Wesley, 1979.

[18] M. Agarwal, A. Sahai, and S. K. Mitter, "Coding into a source: A direct inverse rate-distortion theorem," in *Proceedings of the Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sept. 2006.

[19] ——, "A direct equivalence perspective on the separation theorem," *IEEE Trans. Inform. Theory*, In preparation.

[20] R. Zamir and M. Feder, "On universal quantization by randomized uniform/lattice quantizers," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 428–436, Mar. 1992.

[21] B. M. Leiner and R. M. Gray, "Rate-distortion theory for ergodic sources with side information," *IEEE Trans. Inform. Theory*, vol. 20, no. 5, pp. 672–675, Sept. 1974.

[22] S. Tatikonda, "Control under communication constraints," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[23] B. Rimoldi, "Successive refinement of information: Characterization of the achievable rates," *IEEE Trans. Inform. Theory*, vol. 40, no. 1, pp. 253–259, Jan. 1994.

[24] C. E. Shannon, "The zero error capacity of a noisy channel," *IEEE Trans. Inform. Theory*, vol. 2, no. 3, pp. 8–19, Sept. 1956.

[25] G. D. Forney, "Convolutional codes II. maximum-likelihood decoding," *Information and Control*, vol. 25, no. 3, pp. 222–266, July 1974.

[26] ——, "Convolutional codes III. sequential decoding," *Information and Control*, vol. 25, no. 3, pp. 267–297, July 1974.

[27] F. Jelinek, "Upper bounds on sequential decoding performance parameters," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 227–239, Mar. 1974.

[28] M. S. Pinsker, "Bounds on the probability and of the number of correctable errors for nonblock codes," *Problemy Peredachi Informatsii*, vol. 3, no. 4, pp. 44–55, Oct./Dec. 1967.

[29] L. J. Schulman, "Coding for interactive communication," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1745–1756, Nov. 1996.

[30] A. Sahai and T. Simsek, "On the variable-delay reliability function of discrete memoryless channels with access to noisy feedback," in *Proceedings of the IEEE Workshop on Information Theory*, San Antonio, TX, Nov. 2004.

[31] S. Draper and A. Sahai, "Noisy feedback improves communication reliability," in *Proc. IEEE International Symposium on Information Theory*, July 2006. [Online]. Available: http://www.eecs.berkeley.edu/˜sahai/Papers/DraperSahaiISIT06.pdf

[32] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Trans. Inform. Theory*, vol. 49, no. 3, pp. 563–593, Mar. 2003.

[33] S. Verdu and T. S. Han, "A general formula for channel capacity," *IEEE Trans. Inform. Theory*, vol. 40, no. 4, pp. 1147–1157, July 1994.

[34] S. Tatikonda, A. Sahai, and S. K. Mitter, "Stochastic linear control over a communication channel," *IEEE Trans. Automat. Contr.*, vol. 49, no. 9, pp. 1549–1561, Sept. 2004.

[35] C. Chang and A. Sahai, "The error exponent with delay for lossless source coding," *Information Theory Workshop*, Mar. 2006 to appear. [Online]. Available: http://www.eecs.berkeley.edu/˜sahai/Papers/ITW06.pdf

[36] G. N. Nair and R. J. Evans, "Communication-limited stabilization of linear systems," in *Proceedings of the 39th IEEE Conference on Decision and control*, Sydney, Australia, Dec. 2000, pp. 1005–1010.

[37] ——, "Stabilizability of stochastic linear systems with finite feedback data rates," *SIAM Journal on Control and Optimization*, vol. 43, no. 2, pp. 413–436, July 2004.