Anytime Information Theory: A Summary of Anant Sahai's Ph.D. Thesis

Constantine Caramanis*
Massachusetts Institute of Technology

April 4, 2001

The work presented in this paper is all taken from the Doctoral Thesis of Anant Sahai [1]. In addition, there are some general ideas inspired by the Doctoral Thesis of Sekhar Tatikonda [2], and also by a series of talks given by Professor Mitter (see [3]).

1 Introduction

First we provide a motivation, giving a general description of the high–level ideas behind the material this exposition seeks to develop. Next, we give an outline of the content of this paper.

1.1 Motivation

The classical set–up for the problem of information transmission across a noisy channel, in the context of capacity achieving block codes, has an inherent causal structure, where the transmitter has access to the entire message to be sent at "time -1". However in many practical situations this may not at all be the case. Indeed, the causality relation may be much more complicated, due to some intricate feedback structure in a communications network, or, as will be the case in the situation examined in this exposition, the message may be given only implicitly, and may unfold in time. In this paper we examine the problem of "tracking" with finite distortion, an **unstable** Markov source, across a noisy channel.

In addition to the dynamic specification of the message, there is yet another difference in the problem statement, namely, the ultimate goal is not information transmission in and of itself, but rather the tracking of the unstable Markov source. In some sense, it is this interaction between control, and communication, that is the fundamental idea behind this paper. The causality of the set–up, and the dynamical nature of the message specification, as will be made clear, render the problem sensitive to errors made arbitrarily in the past. Indeed the total end to end distortion plays a crucial role. Because we are interested in this end–to–end performance, there is a conceptual difference from the usual set up. For instance, it is unclear whether or not we can speak of a channel source separation theorem, as is the case with block coding.

In short, the fundamental elements of this paper rest in the interaction between control and information theory, and therefore delay, and distortion sensitivity to delay, as well as the total end–to–end distortion. These issues are all at the core of the material presented, and examining the case of the unstable Markov source illuminates them all. In the words of the author of the Thesis, "For unstable processes, the issues of streaming, delays, and 'real–time' are absolutely essential to the problem of end–to–end communication" ([1]).

*cmcaram@mit.edu

1.2 Outline

The thread that leads us through the material developed in this paper is the problem of tracking an unstable (scalar) Markov source, across a noisy channel, where by tracking we mean achieving a finite expected distortion. This toy problem serves both as an indication that existing ideas are in some sense insufficient, but also to motivate the need and appropriateness of the new material introduced.

In this paper, we introduce a new notion of channel capacity, that turns out to be stronger than the classical Shannon capacity, yet weaker than Shannon's "zero–error" capacity. In fact, we introduce a parametric family of channel capacities, which in one direction, the limit of the parameter yields classical Shannon capacity, and in the other, the limit yields zero error capacity. This parameter has a quality of service interpretation, as discussed in section 5.

In section 2, we introduce the major problem motivating this paper, and in addition, we illustrate that current notions of capacity, and block coding, are inadequate, with respect to the natural performance measure induced by our problem. In section 3, we develop the new parametrized family of capacities, called "Anytime Capacity." Next we show that the notion of "anytime capacity" is meaningful, in the sense that it is indeed stronger than zero error capacity and weaker than Shannon capacity, in a host of interesting cases. That is, we show for a fairly wide class of channels, that anytime capacity is below Shannon capacity, yet strictly greater than zero error capacity (which, in most interesting cases, is zero). Having shown that anytime capacity is in some sense meaningful, we go on in section 4 to show that it is actually useful. We use anytime capacity, along with appropriate encoders and decoders, to solve the problem posed in section 2. Furthermore, we provide a converse statement, which shows that anytime capacity is in fact *exactly* the right thing to consider. Finally, in section 5, we discuss the quality of service interpretation of the anytime parameter, and in section 6 we conclude.

2 The Problem

In this section we introduce the problem of tracking an unstable Markov source. Before we formulate the general statement of the problem, we give some definitions.

Definition 1 *Given a distortion measure ρ, we say that a random process $\hat{X}_t$ ρ–tracks another random process X_t, if the expected distortion is always bounded, i.e.*

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

For the purposes of this paper, we restrict our attention to squared error distortion:

$$\rho(X_t, \hat{X}_t) = (X_t - \hat{X}_t)^2.$$

Furthermore, in all that follows, by "noisy channel" we mean a discrete time channel with 0 delay, and unit sampling time, that is a random map from the input alphabet $\mathcal{A}$, to the output alphabet $\mathcal{B}$, neither of which need be discrete. Then we have,

The Problem: *Let X_t be a Markov process with parameter $A > 1$, driven by noise $\{W_t\}$:*

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

Is it possible to transmit information across a noisy channel, so that the receiver can construct a process $\hat{X}_t$ that ρ-tracks the process X_t?

Note that the dual problem to this is the rate distortion formulation, namely, can we track an unstable Markov source, as in the above statement, across a noiseless channel with finite rate.

2.1 Usual Methods are Insufficient

First, it is clear that the most naive attempt at tracking, namely, to encode and send the actual state of the source X_t, cannot work when we restrict ourselves to any finite rate. Consider the simplified case of a random walk, i.e. $A = 1$, and W_t is a zero mean binary random variable. By the instability of the source (since $A \geq 1$) the process X_t will assume arbitrarily large values with probability 1. A finite rate code can only encode finitely many values. Therefore this naive attempt is clearly inadequate. Intuition tells us immediately that there must be a better way, for over a noiseless channel, all we need to do is transmit the single bit representing the outcome of the random noise, and we can track the random walk perfectly. This is true for $A \geq 1$ as well. However the assumption of perfect transmission is crucial to this solution. In fact, it is insurmountable. This follows because in the proposed tracking method, the receiver's estimate $\hat{X}_t$ *never* loses sensitivity to bit errors made in the past. In fact, if $A > 1$, the squared distortion can grow exponentially as A^{2n} for bit errors that occurred n time units in the past. Even for $A = 1$ we are in trouble. For consider any block code approach, where we can encode the outcome of the random noise W_t with arbitrarily low probability of error. The point is that while the probability of error may be arbitrarily low, it is nonetheless nonzero. Therefore there is a nonzero probability or error for any given bit. By the memoryless nature of the errors (namely, the errors in decoding block k are independent of the errors made in decoding block $k+l$) the difference of the two processes X_t and $\hat{X}_t$, will behave like a random walk, and thus will attain arbitrarily large values with probability one. Therefore we have (loosely) shown the following

Proposition 1 *Traditional block coding arguments are inadequate to track unstable Markov sources, as in the problem statement above.*

If block coding does not meet our needs, after asking "what will," a second, yet equally important question is to ask what happens of our source channel separation theorems. We come to this in section 4.

2.2 Bit Sensitivity

There is an important distinction that has been implicitly drawn above between block codes, and the tracking of the Markov process. Block codes have the property that the dependence on errors made in the past eventually completely disappears. In fact, this happens as soon as we begin decoding the next block. In other words, if we are not transmitting a Markov process, but rather just codewords, $X_1, X_2, X_3, \ldots$, then if we use traditional block codes to transmit the codewords across the channel, then $\rho(X_t, \hat{X}_t)$ will be independent of any (and all) errors made in decoding X_k, for all $k < t$. Therefore, if the source is stationary,

$$\sup_{t>0} E[\rho(X_t, \hat{X}_t)] = E[\rho(X_1, \hat{X}_1)] < \infty,$$

if our channel has a sufficient capacity.

On the other hand, as we have seen, in the case of an unstable Markov source, not only is this sensitivity never lost, but to the contrary, it may grow exponentially. Something qualitatively different is needed. The first question then is, what channel classification is necessary in order to render this problem tractable? Is a nonzero classical Shannon capacity sufficient? Is a nonzero zero–error capacity necessary? To this question, we now turn.

3 Any Time Capacity

In this section we define a parametrized family of capacities, thus introducing a stronger notion of reliable transmission. We establish the relation with classical Shannon capacity, and then we show that this new notion is meaningful, with respect to a fairly wide class of channels. The fundamental idea behind anytime capacity, and the associated encoders and decoders, is that we have the ability to actually *correct* past errors. Evidently (yet loosely), from the informal discussion above, it is something along these lines that is necessary for tracking an unstable source, if we are to allow our channel to produce any transmission errors.

3.1 The Definition

First we define what we mean by an anytime decoder for a channel with unit sampling time, and relative offset θ of the enterring bit stream. We denote by $\mathcal{B}^i$ the possible sequences of outputs up to time i.

Definition 2 *For θ as above, a rate $R > 0$ bits per unit time anytime channel decoder is a sequence of maps, $\mathcal{D}^a = \{\mathcal{D}_i^a\}$,*

$$\mathcal{D}_i^a : \mathcal{B}^i \longrightarrow \{0,1\}^{\lfloor(\theta+i)R\rfloor}.$$

Note that these decoders do not produce a single output stream, but rather, at each instant, they give their best estimate of all past bits as well. The trick will be to find a way for these estimates to continuously improve, therefore correcting bits at the output. The receiver who intends to use, in some way, the output bits, must then choose how many chances the anytime decoder may have to decode a particular bit correctly. While in principle the decoder may wait a different time for each bit, for our purposes here, we assume that a uniform delay d is chosen for all bits, and thus we generate an output sequence

$$\hat{s}_i = (\mathcal{D}_j^a(b_1^j))_i,$$

where

$$j = \lceil \frac{(i-\theta)}{R} + d \rceil.$$

Definition 3 *Given an encoder $\mathcal{E}$, the maximum likelihood anytime decoder $\mathcal{D}^a$, is given by*

$$\mathcal{D}_i^a(b_1^i) = argmax_{s_1^{N_i} \in \{0,1\}^{N_i}} P(B_1^i | A_1^i = \mathcal{E}(s_1^{N_i})),$$

where $N_i = \lfloor(\theta + i)R\rfloor$.

To the extent that we encode a stream of bits so that future bits may provide additional information about past bits, this decoder will be able to correct past errors. Note then, that if indeed past errors get corrected at some rate, then the probability of error will be decreasing in the delay d which the receiver waits before using a particular estimate of a bit from the decoder's output. It is precisely this decay of the probability of error in which we will be interested. It is plausible to imagine that to encode in such a way that the decay speed is greater, requires more redundancy and thus a lower rate of information transmission. It is exactly this tradeoff that the parametric family of capacities captures.

Now, let $f(d) > 0$ be any decreasing function of delay. Then we have the following key definition.

Definition 4 *The f–anytime capacity $C_{anytime}(f) = C_a(f)$ of a channel is the supremum of rates at which the channel can transmit data, such that the probability of error is arbitrarily small, and moreover decays at least as fast as the function f does. In symbols,*

$$C_a(f) = \sup\{R \,|\, \exists\, K > 0,\ Rate(\mathcal{E}, \mathcal{D}^a) = R, \forall\, d > 0,\ P_e(\mathcal{E}, \mathcal{D}^a, d) \leq K f(d)\}.$$

We specialize this definition to the case of exponential functions $f(d) = 2^{-\alpha d}$.

Definition 5 *The α–anytime capacity $C_a(\alpha)$ of a channel, is defined using the above definition, and*

$$C_a(\alpha) = C_a(2^{-\alpha d}).$$

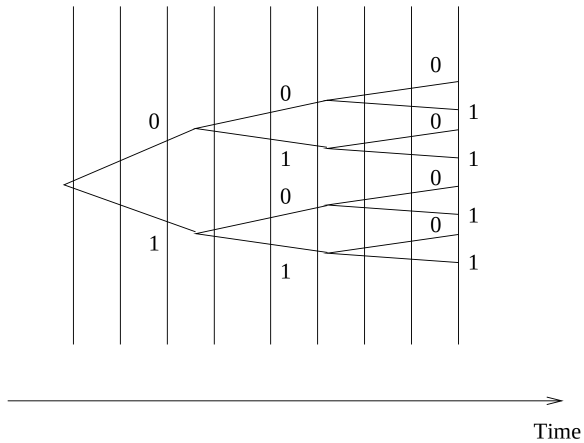

Figure 1: Random Encoder

As mentioned above, this introduces a stronger sense of reliable communication, for rather than merely requiring an arbitrarily low probability of error, we insist that it also go to zero at some rate. The next result shows that if this error decay rate is sufficiently fast, then all bits are eventually decoded correctly, with probability one.

Proposition 2 *If $f(d)$ is summable, i.e. $\sum_{d=1}^{\infty} f(d) < \infty$, then there exists an encoder/decoder pair such that for any i, the decoder's estimate for the i^{th} bit converges to the correct value.*

Proof. This can essentially be viewed as an application of the Borel-Cantelli lemma. □

In section 3.2 below, we show that the binary erasure, and AWGN channels have nonzero α–anytime capacity, and we also cite the general result from [1] which lower bounds the anytime capacity away from zero, for a general memoryless channel. First, however, it is clear that we have, for any $\alpha > 0$,

$$C_{\text{zero–error}} \leq C_{\text{anytime}}(\alpha) \leq C_{\text{Shannon}},$$

and in general all inequalities are strict. The first inequality becomes an equality when $\alpha \to \infty$, and the second when $\alpha \to 0$.

3.2 A Lower Bound on Anytime Capacity

In this section, we use random encoders. Notice in the following definition that unlike in the usual definition of block codes, the dependence on past bits never disappears. As above, we define $N_i := \lfloor(\theta + i)R\rfloor$.

Definition 6 *A random rate R offset θ encoder $\mathcal{E}$ generated according to a distribution $P(a)$, generates the channel inputs $a_i = \mathcal{E}_i(s_1^{N_i})$ such that each a_i is drawn according to the distribution $P(a)$ on the channel input alphabet $\mathcal{A}$. $\mathcal{E}_i(x_1^{N_i})$ and $\mathcal{E}_j(y_1^{N_j})$ are independent whenever $i \neq j$, or $x_1^{N_i} \neq y_1^{N_j}$.*

We illustrate this definition in figure 1, for a rate 1/3 encoder. The input bits navigate our way through the binary tree, and the encoder outputs the realizations of *iid* random variables which can be thought of sitting on each leg of each branch of the tree. Since all the random variables sitting on this tree are *iid*, it is clear that the code generated has the property of independence asserted at the end of the definition above.

3.2.1 Binary Erasure Channel

We give this example first, because while it is rather simple, it gives a good insight into the general case. Throughout this section then, we consider a binary erasure channel with probability e of erasure. Given a particular output of bits b_1^i from the channel, we call a potential sequence of bits *incompatible* with b_1^i if the probability that they were the actual input bits is zero. Note that once a sequence $\tilde{s}_1^i$ is labeled as incompatible with the output of the channel, then so are all extensions $\tilde{s}_1^{i+k}$ of that sequence. Note further that we can decode correctly if there is a unique sequence that is not incompatible with the channel output. We facilitate a precise analysis with this intuition by the following.

Definition 7 *Given a sequence channel outputs b_1^i, the set of j–long binary strings not incompatible with b_1^i are denoted $\mathcal{C}_j^i(b_1^i)$.*

Therefore we have $\mathcal{C}_j^{i+l}(b_1^{i+l}) \subseteq \mathcal{C}_j^i(b_1^i)$. Furthermore, the correct sequence is never incompatible with the output of the channel, so the $\mathcal{C}_j^i$ can never be empty. Therefore the limiting set $\mathcal{C}_j^\infty(b_1^\infty)$ exists, and if it contains a single element, that element must be the correct input sequence. We will use this to prove the main result of this subsection.

Proposition 3 *For the binary erasure channel with probability of erasure e, we have*

$$C_{anytime}(\alpha) \geq 1 - \log_2(1+e) - \alpha,$$

and moreover, this lower bound is achievable.

In particular then, $C_a(\alpha) > 0$ whenever $\alpha < 1 - \log_2(1+e)$. We prove two lemmas which will together yield the proof of the theorem. First we relate the probability of a possible decoding ambiguity, to the probability that the first ambiguity is introduced at a particular stage.

Lemma 1 *For $0 < \lfloor(\theta + j)R\rfloor \leq i$,*

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

where we set $\mathcal{C}_j^i = \{\emptyset\}$.

Proof. The proof is a straightforward computation.

$$\begin{aligned}
P(||\mathcal{C}_j^i(B_1^i)|| > 1) &= P(||\mathcal{C}_j^i(B_1^i)|| > 1 \Big| ||\mathcal{C}_{j-1}^i(B_1^i)|| = 1) P(||\mathcal{C}_{j-1}^i(B_1^i)|| = 1) \\
&\quad + P(||\mathcal{C}_j^i(B_1^i)|| > 1 \Big| ||\mathcal{C}_{j-1}^i(B_1^i)|| > 1) P(||\mathcal{C}_{j-1}^i(B_1^i)|| > 1) \\
&= P(||\mathcal{C}_j^i(B_1^i)|| > 1 \Big| ||\mathcal{C}_{j-1}^i(B_1^i)|| = 1) P(||\mathcal{C}_{j-1}^i(B_1^i)|| = 1) + P(||\mathcal{C}_{j-1}^i(B_1^i)|| > 1) \\
&= \sum_{k=1}^{j} P(||\mathcal{C}_{j-k+1}^i(B_1^i)|| > 1 \Big| ||\mathcal{C}_{j-k}^i(B_1^i)|| = 1) P(||\mathcal{C}_{j-k}^i(B_1^i)|| = 1) \\
&\leq \sum_{k=1}^{j} P(||\mathcal{C}_{j-k+1}^i(B_1^i)|| > 1 \Big| ||\mathcal{C}_{j-k}^i(B_1^i)|| = 1) \\
&\leq \sum_{k=1}^{\infty} P(||\mathcal{C}_{j-k+1}^i(B_1^i)|| > 1 \Big| ||\mathcal{C}_{j-k}^i(B_1^i)|| = 1).
\end{aligned}$$

□

The next result gives a bound on the probabilities making up the right hand side of the above bound.

Lemma 2 *If the input stream S_1^∞ is generated by iid Bernoulli(1/2) random variables, then for the above channel,*

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

PROOF. The key part of this proof comes from the manner in which we defined our random encoders. In what follows, we let s_1^∞ denote the true input sequence of bits. Now, the event $\{||\mathcal{C}^i_{j-k+1}(B_1^i)|| > 1 \Big| ||\mathcal{C}^i_{j-k}(B_1^i)|| = 1)\}$ can have positive probability only if we can find bits to complete the sequence $(s_1, \ldots, s_{j-k}, 1-s_{j-k+1}, \tilde{s}_{j-k+2}, \ldots, \tilde{s}_{\lfloor(\theta+i)R\rfloor})$ to a sequence not incompatible with B_1^i. The first channel bit that depends on the bit $1 - s_{j-k+1}$, is $\left\lceil \frac{j-k+1-\theta}{R} \right\rceil =: M_{jk}$. Let $\tilde{A}^l = (\tilde{A}_{M_{jk}}, \ldots, \tilde{A}_i)$ be a possible string that could have resulted in the same channel output as the true string A^l. We have $l = i - M_{jk}$. By definition of our random encoder, we must have that $\tilde{A}^l$ is independent from the true transmitted sequence A^l. The probability of a randomly generated length l string being indistinguishable from A^l is

$$\begin{aligned} P(\tilde{A}^l, A^l \text{ indistinguishable}) &= \sum_{m=0}^{l} \frac{2^{-m} l! e^{l-m}(1-e)^m}{m!(l-m)!} \\ &= \left(\frac{1+e}{2}\right)^l, \end{aligned}$$

by the binomial expansion.

Now, the l channel uses can correspond to at most $1 + lR$ input bits, and hence at most 2^{1+lR} input sequences. Therefore we can bound the probability of the first ambiguity appearing at stage $j - k + 1$ by

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

as desired. □

With the above results, we can give the proof of proposition 3 above.

PROOF. We know that there can be a decoding error in bit j at time i, if and only if the set $\mathcal{C}^i_j(B_1^i)$ is not a singleton. Therefore we have,

$$\begin{aligned} P(\hat{S}_j \neq S_j)) &\leq P(||\mathcal{C}^i_j(B_1^i)|| > 1) \\ &\leq \sum_{k=1}^{\infty} P(||\mathcal{C}_{j-k+1}(B_1^i)|| > 1 \Big| ||\mathcal{C}^i_{j-k}(B_1^i)|| = 1) \\ &< \sum_{k=1}^{\infty} 2\left(\frac{1+e}{2^{1-R}}\right)^{i-\left\lceil \frac{j-k+1-\theta}{R} \right\rceil + 1} \\ &< 2 \sum_{l=i-\left\lceil \frac{j-\theta}{R} \right\rceil} \left(\frac{1+e}{2^{1-R}}\right)^l \\ &= \frac{2\left(\frac{1+e}{2^{1-R}}\right)^{i-\left\lceil \frac{j-\theta}{R} \right\rceil}}{1 - \frac{1+e}{2^{1-R}}} \\ &\leq \frac{2\left(\frac{1+e}{2^{1-R}}\right)^{-1+i-\frac{j-\theta}{R}}}{1 - \frac{1+e}{2^{1-R}}}. \end{aligned}$$

Now, if $R \leq 1 - \log_2(1 + e)$, then the probability of error of the j^{th} bit at time i is an exponentially decaying function of the delay $i - \frac{j-\theta}{R}$. But then writing

$$\left(\frac{1+e}{2^{1-R}}\right) = 2^{-\alpha},$$

and solving, we obtain $R = 1 - \log_2(1 + e) - \alpha$, as claimed. □

3.2.2 Additive White Gaussian Channel

The encoder used for this channel will be much the same as the one used in the binary erasure channel. The same tree is formed, yet rather than drawing *iid* binary random variables, we draw *iid* zero mean Gaussian random variables, with variance P (thus satisfying the channel's power constraint). In this case, we use the density to obtain a maximum likelihood decoder. Again letting $N_i = \lfloor(\theta + i)R\rfloor$, we have,

$$\begin{aligned}
\mathcal{D}_i^a(b_1^i) &= \text{argmax}_{s_1^{N_i} \in \{0,1\}^{N_i}} p(B_1^i = b_1^i | A_1^i = \mathcal{E}(s_1^{N_j}) \\
&= \text{argmax}_{s_1^{N_i} \in \{0,1\}^{N_i}} \frac{1}{i} \sum_{j=1}^{i} (b_j - \mathcal{E}_j(s_1^{N_j}))^2.
\end{aligned}$$

By the law of large numbers, the cost of the true path will be the variance of the channel — that is, 1, while any false path will have average cost that tends to $1 + 2P$. With the help of the following two lemmas, analogs of lemma 1, and lemma 2 above, we prove the main result of this subsection, a lower bound to the AWGN channel anytime capacity.

Proposition 4 *The power constrained AWGN channel has*

$$C_{anytime}(\alpha) \geq \frac{1}{2}\log_2(1 + \frac{P}{2}) - \alpha,$$

and this bound is achievable.

We now state and prove the two necessary lemmas.

Lemma 3 *For* $0 < \lfloor(\theta + j)R\rfloor \leq i$,

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

PROOF. By induction, we have the following sequence of inequalities:

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

□

Next we have,

Lemma 4 *If our noisy channel is an AWGN channel with power constraint P, and if the input stream S_1^∞, is, as before, generated by an iid Bernoulli(1/2) process, then*

$$P((\mathcal{D}_i^a(B_1^i))_{j-k+1} \neq s_{j-k+1} \Big| (\mathcal{D}_i^a(B_1^i))_{-\infty}^{j-k} = s_{-infty}^{j-k}) \leq 2\left(\frac{2^R}{\sqrt{1+\frac{P}{2}}}\right)^{i - \lceil \frac{j-k+1-\theta}{R} \rceil + 1}.$$

PROOF. The proof is quite similar to lemma 2, but notationally heavy, so we omit it here, but see [1]. □

Now we can prove proposition 4 above.

PROOF. The proof is quite similar to the proof for the binary erasure channel analog. We have, from the two lemmas above,

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

If $R < \frac{1}{2}\log_2(1+\frac{P}{2})$, we have exponential convergence of the probability of error. Solving, as before, we find that $R = \frac{1}{2}\log_2(1+\frac{P}{2}) - \alpha$, as claimed. □

3.2.3 The General Case

We state, but do not prove, the general statement (but, as for the proof of all the results here, see [1]).

Theorem 1 *For a memoryless channel with block random coding exponent $E_r(R)$, we have*

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

where $E_r(R)$ is Gallager's standard block coding exponent (see [4]) and $e = 2.71828\ldots$.

Note that all of our statements have been made and then proved for random encoders, that is to say, encoders/decoders that share common randomness. We can think of our results on performance of random encoders to be the expected performance of deterministic encoders with respect to the measure generated by the randomness used above. Recall that the usual achievability proofs, e.g. in [5], work in much the same way, and then argue that there must exist a deterministic encoder that achieves the given bounds. In our case, however, we are considering an expectation over an *uncountable* number of infinite encoders, so showing the existence of a deterministic encoder with desired performance is no longer as simple. See [1] for further details.

4 The Solution to the Problem

In this section we use the notion of anytime capacity introduced in section 3 above, to solve the problem of tracking the unstable source. We show first, that if a noisy channel has a sufficiently large α–anytime capacity for a sufficiently large α, then we can track a scalar unstable Markov process across the channel. Conversely, we show that this is exactly what we need, by showing that if we can track an unstable Markov source across a noisy channel, then the channel has appropriate α–anytime capacity.

4.1 The Direct Part

In section 2 of this paper, we saw that block codes are inadequate to address the problem of tracking an unstable Markov source. The primary reason for this is that the effect of errors is at the very least cumulative, rather than dying off with time. In section 3 above, we developed encoders that create codes whose dependence in time does not die off. Therefore while rendering the decoding sensitive to past errors, nevertheless this dependence allows for the possibility of *correcting* past errors. In this section we show that if a channel has sufficient α–anytime capacity, for a sufficiently large α, then it can track an unstable source. We expect that the amount of anytime capacity we require is related to the exponential growth rate of the unstable source. Indeed this turns out to be the case. To see more precisely why this might be the case, we introduce more formally the idea of dependence on past errors. Letting ρ be our distortion measure, we define $\Delta^+(j,t), \Delta^-(j,t)$, the greatest distortion *at* t that can be caused by mistakes made from time j on, and the greatest distortion *at* t that can be caused by mistakes made up to time j, respectively. In what follows, we denote by F our encoder, and by $\hat{x}_t$ the receiver which reassembles the source X_t.

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

Also, we have

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

Note that for any code (such as block codes) which loses dependence on past errors, $\Delta^+(d)$ does not depend on d for $d > N$, some N. The next theorem shows that this is one of the fatal flaws of block codes, with respect to tracking the unstable source, and that indeed, the problem rests not only with block codes, but in fact with any code with a decaying sensitivity to past errors.

Theorem 2 *For an unstable Markov source, as above, and any finite expected squared-error-distortion code, $\Delta^+(d)$ grows exponentially at least as fast as A^{2d}, with A the parameter of the source.*

For the proof, and a much more involved discussion of Δ^+ and Δ^-, and distortion, again, see [1], chapters 3 and 4. Finally we can state the direct part of the main theorem.

Theorem 3 *Consider a source code with finite expected distortion D_R, that emits a rate R stream of bits. Suppose our noisy channel has $C_a(f) > R$ for a function $f(d)$ that decays quickly enough so that*

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

Then, we can trasmit the source code across the noisy channel, with finite expected distortion that can be made to approach D_R arbitrarily (as long as we are willing to accept sufficient additional end–to–end delay δ').

Proof. Let $\delta' \geq 0$ be the additional delay we choose to tolerate. Because we have sufficient anytime capacity, by the results of section 3, we know that the initial bits will eventually be correctly retrieved. We then have our estimates of the source,

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

We bound our distortion by

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

from which point the result quickly follows. □

Now by Theorem 2, the following Theorem, solving our problem from section 2, follows as a corollary to the above.

Theorem 4 *A scalar discrete time unstable Markov process as given above, driven by bounded noise, can be tracked with finite squared–error–distortion across any noisy channel such that there is some* $\varepsilon > 0$ *for which* $C_a(2\log_2 A + \varepsilon) > \log_2 A$*, for the channel.*

Besides providing a direct solution to our problem, theorem 3 above also provides the answer to another question we posed earlier: The question of source channel separation. Indeed theorem 3 says that if our aim is to encode a source to some distortion D_R, and then transmit it across a noisy channel, we can optimally encode our source, and then cascade it with an anytime encoder, as described above. Chapter 4 of [1] describes in detail a source coding problem, and its solution, again in terms of codes which do not lose sensitivity to bit errors, in the sense of Δ^+ and Δ^- introduced above. It is clear from that discussion that if our performance measure is end–to–end distortion, a scheme such as the one described is optimal. However, evidently, we cannot cascade such a code with a typical block channel coder, as the latter produces a so–called strong stream of bits (see [1], chapter 3) and therefore does not respect the enduring dependencies which are necessary for tracking the unstable source across the noisy channel, or over a rate constricted channel. Theorem 3 comes to the rescue, however, and proves that in this case as well, we have a type of channel source separation.

4.1.1 The Converse Direction

In this section we state the theorem that says that in fact, we *need* to have a sufficient anytime capacity with sufficiently large α, in order to track an unstable Markov source. The proof of the necessity follows a different direction than the usual converse arguments follow. We show by construction that if we can track an unstable Markov source across a noisy channel, then that noisy channel must have a certain anytime capacity. We show this by actually constructing the anytime encoders and decoders.

In this section we give the main idea behind the proof, but also provide most of the details, leaving only a few, conceptually straightforward, yet notationally tedious details to [1]. The fundamental idea is to take the input sequence S_1^∞, and construct a simulation of an unstable source. Then from the finite–distortion tracking of this source at the receiver, we try to recover the original bit sequence with probability of error decaying at the appropriate rate. Before we discuss the ideas behind the proof, we give the main statment:

Theorem 5 *Suppose we can track any scalar unstable Markov process X_t with parameter $A > 1$, driven by a bounded noise signal W_t, with $||W_t|| \leq \frac{\Omega}{2}$, across a given channel. Specifically, suppose that for this channel we have*

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

Then, $\forall \varepsilon > 0$, this channel has

$$C_{anytime}(\tilde{f}) > \log_2 A - \varepsilon,$$

where $\tilde{f} := f(\delta' 2^{d \log_2 A})$, for some $\delta' > 0$.

Note that in particular, this theorem tells us that if we can η–track any process of the class above, then we can take $\tilde{f}$ to be exponential, and therefore $C_a(\eta \log_2 A) > \log_2 A - \varepsilon$.

The fundamental idea lies in finding a clever representation of the real numbers $\mathbb{R}$ in a way that decoding the first bits will somehow be more reliable than the subsequent bits. As mentioned, we want to simulate an unstable source. The easiest way to do this is to let W be noise taking two values, and then let the incoming bits specify the "realizations" of the random variable. We will simulate a source $\tilde{X}_t$ by

$$\tilde{X}_t = \sum_{i=1}^{t} A^{t-i} W_i.$$

Now, take $A = 2 + \varepsilon_1 = 2^{1+\varepsilon_2}$. Consider a 1–bit noise signal W_t taking values in $\{-\delta/2, \delta/2\}$, depending on whether $S_t = 0$, or 1, respectively. We will satisfy the theorem if we can find a rate $1 = \log_2 A - \varepsilon_2$ encoder for arbitrary $\varepsilon_2 > 0$. Then if we let S_1^∞ denote our input stream of bits, we have

$$\tilde{X}_t = \sum_{i=1}^{t} A^{t-i} W_i = \frac{\delta}{2} \sum_{i=1}^{t} (2 + \varepsilon_1)^{t-i} (2S_i - 1).$$

Note that because $A > 1$, the "earlier" values of W play an exponentially more prominent role in the value of our constructed process $\tilde{X}_t$. The key in the proof takes advantage of exactly this fact. The proof defines a new representation of real numbers as infinite sequences of zeros and ones. This is very similar in spirit to the binary or ternary representations with which we are familiar. We now briefly review the important points of these representations.

Recall that for any $\beta \in [0, 1]$, we can write

$$\beta = \sum_{n=1}^{\infty} a_n 2^{-n},$$

for $a_i \in \{0, 1\}$. Given β, we can compute its binary expansion by a series of translations and comparisons. At stage 1 of the algorithm, set $x = \beta$. Then at stage n of the algorithm, we compare x to 2^{-n}. If $x < 2^{-n}$, then we set $a_n = 0$, otherwise we set $a_n = 1$. Finally, we update n, and set $x = x - a_n 2^{-n}$, and repeat the procedure. Note that if $\beta_1, \beta_2 \in [0, 1]$ have binary expansions differing for the first time on the k^{th} digit, then we can upper bound their difference,

$$|\beta_1 - \beta_2| \leq 2^{-k+1}.$$

Note, however, that we cannot also provide a lower bound.

Suppose next that we interpret the infinite binary sequences above as ternary sequences, i.e. a sequence $\{a_n\}_{n=1}^{\infty} \in \{0, 1\}^{\infty}$ corresponds to the real number

$$\gamma = \sum_{n=1}^{\infty} a_n 3^{-n}.$$

Then the infinite sequences correspond to the standard Cantor set, scaled by a factor of $\frac{1}{2}$. Given an element of the Cantor set, we can retrieve its expansion in precisely the same manner of translations and threshold comparisons outlined above, but this time using thresholds 3^{-n} (note that we must now *a priori* that our given real number lies in the Cantor set). There is an interesting expression of the fact that this set has lebesgue measure zero, while the set of binary expansions has lebesgue measure one. Suppose, as we did above, that the expansions for β_1, β_2, two elements of the Cantor set, differ for the first time in the k^{th} digit. Call these expansions $\{a_n\}$, and $\{b_n\}$ respectively. Then while we can provide an upper bound to the difference:

$$|\beta_1 - \beta_2| \leq 3^{-k+1},$$

we can *also* provide a lower bound. Suppose, without loss of generality, that $a_k = 1$, and $b_k = 0$. Then,

$$\begin{aligned} |\beta_1 - \beta_2| &= \sum_{i=1}^{\infty}(a_i - b_i)3^{-i} \\ &= 3^{-k} + 3^{-k}\sum_{i=1}^{\infty}(a_{k+i} - b_{k+i})3^{-i} \\ &\geq 3^{-k} - 3^{-k}\sum_{i=1}^{\infty}3^{-i} \\ &= \frac{1}{2}3^{-k}. \end{aligned}$$

In a sense then, this is a converse statement, that says that if β_1, β_2 are sufficiently close in the usual metric on $\mathbb{R}$, then their binary representations agree for the first certain number of bits. Note where the above breaks down in the case of the binary expansion.

In this proof, we encounter a generalization of the standard Cantor set given above. Given an element of this set, again we retrieve the bits of the binary representation by a generalization of the method of comparisons to thresholds, followed by appropriate translations. Essentially, we construct a simulated unstable Markov source using our input sequence of bits, then track it across the channel, and then attempt to recover the input bits via a theshold comparison method, such as the one outlined above. Specifically, given a sequence of thresholds $\{T_n\}$, and offsets $\{O_n\}$, and given $x \in \mathbb{R}$, we obtain an infinite string s_1^∞ by comparing x to threshold T_n at time n, concatenating a 1 or 0, as appropriate to our string, translating x by O_n, and repeating the procedure updating n. The important point on which we must focus is how the distortion of the tracking affects the bit retrieval.

Now we consider the mapping proposed above, that constructs a simulation of an unstable Markov source, by letting the input bits give the "realization" of the noise:

$$\tilde{X}_t = \frac{\delta}{2}\sum_{i=1}^{t}(2+\varepsilon_1)^{t-i}(2S_i - 1).$$

By appropriate translations and normalizations (all invertible, see below) we obtain the new mapping

$$\tilde{X}'_t = \sum_{i=1}^{t}(2+\varepsilon_1)^{-i}S_i.$$

Now, consider the mapping from the space of binary sequences to the real numbers, given by

$$\tilde{x}'_\infty(s_1^\infty) := \sum_{i=1}^{\infty}(2+\varepsilon_1)^{-i}s_i.$$

Notice that this mapping has a range space $\mathcal{X}$ that is very similar to that of the Cantor set. $\mathcal{X}$ is in fact a "Cantor set," and if we were to take $\varepsilon_1 = 1$, it would be the standard Cantor set discussed above. It is quite easy to see, in particular, that when $\varepsilon_1 > 0$, $\mathcal{X}$ has zero lebesgue measure. This is very important, because of the lower bound property shown above. Given the analogy to the familiar Cantor set, the results of the next lemma are almost clear, but due to their importance, we state them.

Lemma 5 *If $\varepsilon_1 > 0$, then the mapping above is strictly monotonic with respect to the usual ordering on $\mathbb{R}$, and lexicographic ordering of the sequences. Furthermore, if s_1^∞ and $\hat{s}_1^\infty$ first differ in digit n, then*

$$|\tilde{x}'_\infty(s_1^\infty) - \tilde{x}'_\infty(\hat{s}_1^\infty)| \geq (2+\varepsilon_1)^{-n}\frac{\varepsilon_1}{1+\varepsilon_1} > 0.$$

Note that $\varepsilon_1 > 0$ is critically important, for otherwise, as is apparent in the lower bound itself, the lower bound becomes trivial.

Proof. The proof is a long series of straightforward, although tedious manipulations, similar to the notationally easier ones performed above. □

We can use the result of the above lemma to find thresholds and shifts which will allow us to decode the bits encoded into the simulation of the unstable source.

Lemma 6 *For the mapping $\tilde{x}$ given above, we can take offsets*

$$O_n = (2+\varepsilon_1)^{-n},$$

and any thresholds

$$(2+\varepsilon_1)^{-n} \geq T_n > (2+\varepsilon_1)^{-n}(1 - \frac{\varepsilon_1}{1+\varepsilon_1}),$$

and thus recover the bits exactly (when no noise is introduced).

Proof. Again, we do not go into the details of the proof, as they are simply a more tedious verification of results which are clear in the notationaly easier case. □

Now comes the key. If we attempt to apply the exact recovery procedure proposed in the lemma above, but in the case when we *do* have noise from the channel, what can we say? In other words, does this all break down if we run the above procedure on $X' = X + \zeta$? The previous lemma tells us that depending on the magnitude of ζ, we can be sure to retrieve the first several bits correctly. Specifically, if

$$|\zeta| < \frac{1}{2}(2+\varepsilon_1)^{-k}\frac{\varepsilon_1}{1+\varepsilon_1},$$

then we correctly determine the first k bits. From here we use the fact that we can track the source to finite distortion to bound the probability that the channel distortion magnitude $|\zeta|$ above, exceeds the bound which guarantees correct decoding of the first k bits. This noise is a scaled version of the difference $|X_t - \hat{X}_t|$, of the original process and the process the receiver recreates. By the assumption of the theorem, we have bounds on this difference, as it is simply the distortion.

We perform the decoding, and the error computation as follows. First, we translate and normalize the recreated process to obtain

$$x' = \frac{1}{\delta A^t}\hat{X}_t + \frac{1+(2+\varepsilon_1)^{-t}}{2+2\varepsilon_1}.$$

Next, we apply the threshold procedure outlined above, to obtain the first t bits. By the lemma above, we know that if the distortion is sufficiently small, then it will not be expressed in the first k bits, i.e. we decode the first k bits correctly. Then, if we can bound the probability of a large distortion, we have also bounded the probability of making an error in the first k bits. Note that this is exactly the connection we need to obtain our anytime capacity. We have,

$$\begin{aligned}
P_e(\mathcal{E}', \mathcal{D}^a, d, k) &\leq P(|\zeta| > \frac{\varepsilon_1}{2+2\varepsilon_1}(2+\varepsilon_1)^{-k}) \\
&= P(|\delta A^{k+d}\zeta| > \frac{\varepsilon_1}{2+2\varepsilon_1}(2+\varepsilon_1)^{-k}\delta A^{k+d}) \\
&= P(|X_t - \hat{X}_t| > \frac{\varepsilon_1}{2+2\varepsilon_1}(2+\varepsilon_1)^{-k}\delta A^{k+d}) \\
&\leq P(|X_t - \hat{X}_t| > \frac{\delta\varepsilon_1}{2+2\varepsilon}(2+\varepsilon_1)^{d}) \\
&\leq f\left(\frac{\delta\varepsilon_1}{2+2\varepsilon}(2+\varepsilon_1)^{d})\right) \\
&= f\left(\frac{\delta\varepsilon_1}{2+2\varepsilon}2^{d\log_2 A}\right).
\end{aligned}$$

Taking $\delta' = \frac{\delta\varepsilon_1}{2+2\varepsilon_1} > 0$, we have the desired result of the theorem. We have proved the theorem for the case $A = 2 + \varepsilon_1$. The case for general $A > 1$ is similar, but requires more care with notation.

5 Quality of Service

In this section we deviate a bit from the direction in which we have so far been moving, in order to illustrate at a high level, the interpretation of the α parameter of anytime capacity as a quality of service measure. For this section we assume that we have a noiseless feedback link. Furthermore, we make use of results in [1], chapter 7, as needed.

The rate of a particular channel can be viewed as a resource, which is additively distributed in some manner, among the "customers" using the channel, as it may be divided among the various demands on the channel by straightforward time sharing. However, since we have feedback at our disposal, there is more we can do than simply time sharing. In other words, there are other ways of differentiating service, besides allocating a larger or smaller fraction of the available rate R to a given user. We illustrate this by means of an extension of the problem we have traced through this paper. Consider a vector unstable Markov process, $X_t \in \mathbb{R}^5$, with

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

with initial condition $X_0 = 0$, where we have bounded noise $||W_t|| \leq \frac{\delta}{2}$, and matrix A given as

$$A = \begin{pmatrix} 1.258 & 0 & 0 & 0 & 0 \\ 0 & 1.058 & 0 & 0 & 0 \\ 0 & 0 & 1.058 & 0 & 0 \\ 0 & 0 & 0 & 1.058 & 0 \\ 0 & 0 & 0 & 0 & 1.058 \end{pmatrix}.$$

Thanks to theorem 4, we know some of the necessary conditions the channel must satisfy. We must have

$$\begin{aligned}
C_a(2\log_2(1.258)) &> \log_2(1.258), \\
C_a(2\log_2(1.058)) &> \log_2(1.058).
\end{aligned}$$

Also from [2] we know that

$$C > \log_2(1.258) + 4\log_2(1.058).$$

From [1] chapter 7, we know that these constraints are individually satisfied by the erasure channel with probability of erasure $e = 0.27$. Is this channel capable of trasmitting and tracking this vector source to finite distortion? The interesting result of this section is that if we consider rate to be the only resource of the channel, and simply chose a way to divide this among the five sources that make up the vector source, then we cannot achieve a finite distortion. The point is, as we will see, that once the coding is complete, the bits cannot be regarded and treated as identical. Because the faster source requires a larger anytime capacity at a larger α, the bits in the source code that correspond to the fast source, must be given preferential treatment by the channel code.

Again drawing from a result in [1] chapter 7, we have

$$\sup\{\alpha : C_a(\alpha) > \log_2(1.258) + 4\log_2(1.058)\} \approx 0.646.$$

The fast component, however, requires $\alpha > 2\log_2(1.258) = 0.662$ in order to be tracked with finite distortion. Therefore this scheme cannot work. Note that in this scheme after the initial coding is complete, the transmission layer treats each bit identically, whether it be derived from the fast, or the slow components.

One way around this problem is differentiated service. We code the sources by allocating $R_1 = 1/3 > \log_2(1.258)$ for the first source, and $R_{2,3,4,5} = 1/12 > \log_2(1.058)$ for the four slow components. Note that this gives $R = R_1 + 4R_{2,3,4,5} = 2/3 < 1 - e = 0.73$. Furthermore, we give the bits corresponding to the encoding of the first stream "priority" over the other bits. We accomplish this by buffering separately the bits from the different components. Then with each channel use, we transmit the oldest bit from the buffer with highest priority. Thus, if there are bits from the fast component, we transmit them, otherwise we transmit bits from the slow components. Since we have noiseless feedback, we know if a bit was received successfully. Each time a bit is received successfully, it is removed from its respective buffer. The analysis in [1] chapter 8 demonstrates that this coding and transmission scheme corresponds to an effective $\alpha = 1.799$ for the fast component, and $\alpha = 0.285$ for the slow components. Since both values are sufficient, the outlined method is fast enough for both the fast, and the slow components. Our source channel separation theorem (theorem 4) then tells us that we can cascade a source code, with this channel code to achieve a finite mean squared error tracking of this vector source.

6 Conclusions

In this paper we have developed and discussed a new notion of reliability in noisy channel communication, called anytime capacity. One of the key motivations behind this development has been the interconnection of control and communication. When there is a value of information in regards to controlling a system, then the paradigm of reliable communication must, as [1] tries to show, be qualitatively different. This is intimately related to issues of delay, and causality, and how they both control, and affect the system at hand.

References

[1] Sahai, A. "Anytime Information Theory." Ph.D. Dissertation, MIT 2001.

[2] Tatikonda, S. "Control Under Communication Constraints." Ph.D. Dissertation, MIT 2000.

[3] Mitter, S. "Control with Limited Information: the role of Systems Theory and Information Theory."

[4] Gallager, R. "Information Theory and Reliable Communication," NY: John Wiley and Sons, 1971.

[5] Cover, T.; Thomas, J, "Elements of Information Theory," NY: John Wiley, 1991.