A Simple Encoding And Decoding Strategy For Stabilization Over Discrete Memoryless Channels

Anant Sahai and Hari Palaiyanur

Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA, 94720-1770
{sahai, hpalaiya}@eecs.berkeley.edu

Abstract

We study stabilization of a discrete-time scalar unstable plant over a noisy communication link, where the plant is perturbed at each time by an unknown, but bounded, additive disturbance. In order to have a simple encoding and decoding strategy, we restrict the observer/encoder to be nearly memoryless in that the channel inputs can only depend on the last sample taken of the plant state, while the controller/decoder is based on the ZJ or Stack Algorithm. Based on an argument due to Jelinek, we show that the suboptimal decoding algorithm achieves stabilization of the same moments as with ML decoding, without the unboundedly increasing computational cost of ML decoding. While in principle the decoder does require an unbounded amount of memory, the computational effort expended at every time instant is a random variable that does not grow with time on average.

1 Introduction

The problem of control in the presence of communication constraints is illustrated in figure 1. This is a problem of cooperative distributed control in that there are two different boxes that need to be designed: the observer/encoder that has access to the plant state and generates an input to the noisy channel, and the controller which has access to the channel outputs and applies a control to the plant. The reader is referred to two resources to get the appropriate background. The first is the classic 1978 paper by Ho, Kastner, and Wong [3] in which they summarize the then known relationships among team theory, signaling, and information theory from the perspective of automatic control. The more recent interest in the subject is best understood by reading the recent September 2004 special issue of Transactions on Automatic Control and the references contained therein.[1]

Here, we focus exclusively on the case shown in figure 1 where the information patterns [9] are non-nested in that the observer does not know everything the controller knows. This means that there is no explicit feedback of the channel outputs to the observer. Our previous work [6] showed that in the case of finite alphabet channels, the plant itself could be used as a "feedback channel" for communication. The controller can make the plant dance so as to noiselessly communicate the channel output to the encoder/observer. This showed that in principle, the non-nested information pattern did not increase the difficulty of the stabilization problem. [6] further showed that the problem of stabilization over a noisy channel was equivalent to a problem of anytime communication using the noisy channel with perfect feedback.

---

[1]Due to space limitations, I only cite references specifically helpful for understanding the present work, not for giving full and proper historical background.
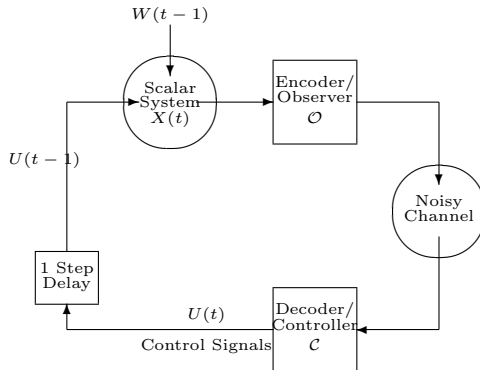
Figure 1: Control over a noisy communication channel without any explicit feedback path from controller to observer.

While this equivalence certainly illuminates the structure of distributed control problems (see [7] for more details), it does not actually allow us to solve any particular problem. This is especially true since the previously known generic constructions for anytime codes are based on infinite random trees. As such, they require growing complexity at both the encoder and decoder. This contrasts in spirit with the idea of stabilization making a system behave in a nice "steady-state" manner that returns to the same few safe states over and over again. If the closed-loop system is behaving nicely, why should the observer and controller have a steadily increasing workload?

In this paper, we attack this problem by making a structural requirement on the observer/encoder as well as the controller/decoder with the intent of keeping the complexity at both encoder and decoded bounded. The observer/encoder must be a nearly memoryless map (though possibly time-varying and random in nature) that picks channel outputs based only on the current state of the plant. The controller/decoder is required to be a sequential decoder as defined in the classic paper of Forney[1].

In section 2, we formally introduce the problems of stabilization and anytime communication and review the existing results from [6], and [7]. In section 3, we adapt the existing strategy by replacing the ML decoder with a sequential decoder. We then state and prove a sufficient condition for stabilization using nearly memoryless random time-varying encoder/observers and sequential decoders over discrete memoryless channels. Next, in section 4, we give simulation results of this strategy and compare them with the analysis. Finally, we comment on the results and in section 5, we look to future enhancements of our scheme.

2 Problem Definition and Review

2.1 Stabilization Problem

At the heart of the stabilization problem is a scalar plant that is unstable in open loop. Expressed in discrete-time, we have:

$$X(t+1) = \lambda X(t) + U(t) + W(t), \quad t \geq 0 \tag{1}$$

where $\{X(t)\}$ is a $I\!R$-valued state process. $\{U(t)\}$ is a $I\!R$-valued control process and $\{W(t)\}$ is a bounded noise/disturbance process s.t. $|W(t)| \leq \frac{\Omega}{2}$ a.s. $\lambda \geq 1$ so the system

is unstable while $X(0) = 0$ for convenience.[2]

The distributed nature of the problem comes from having a noisy communication channel in the feedback loop. We require an observer/encoder system $\mathcal{O}$ to observe $X(t)$ and generate inputs $A(t)$ to the channel. We also require a decoder/controller system $\mathcal{C}$ to observe channel outputs $B(t)$ and generate control signals $U(t)$. In general, we could allow both $\mathcal{O}, \mathcal{C}$ to have arbitrary memory and to be nonlinear. In this paper, we focus on the case of nearly memoryless $\mathcal{O}$.

Definition 2.1 A closed-loop dynamic system with state $X(t)$ is *η-stable* if there exists a constant K s.t. $E[|X(t)|^\eta] \leq K$ for all $t \geq 0$.

Holding the η-moment within bounds is a way of keeping large deviations rare.[3] The larger η is, the more strongly we penalize very large deviations.

2.2 Anytime communication

For simplicity of notation, let M_i be the R bit message that a channel encoder takes in at time i. Based on all the messages received so far, and any additional information (e.g. past channel output feedback) that it might have, the channel encoder emits the i-th channel input. An anytime decoder provides estimates $\hat{M}_i(t)$, the best estimate for message i at time t. If we are considering a delay d, the probability of error we are interested in is $P(\hat{M}_{t-d}(t) \neq M_{t-d})$.

Definition 2.2 The *α-anytime capacity* $C_{\text{anytime}}(\alpha)$ of a channel is the least upper bound of the rates at which the channel can be used to transmit data so that there exists a uniform constant K such that for all d and all times t we have

$$P(\hat{M}_{t-d}(t) \neq M_{t-d}(t)) \leq K 2^{-\alpha d}$$

The probability is taken over the channel and any randomness that we deem the encoder and decoder to have access to. Due to the fast convergence of an exponential, it is equivalent to requiring that the probability of error for *all messages sent upto time* $t - d$ is bounded by $K' 2^{-\alpha d}$.

2.3 Anytime communication and stabilization

Theorem 2.3 [6] For a given noisy channel, bound Ω, and $\eta > 0$, if there exists an observer/encoder $\mathcal{O}$ and controller/decoder $\mathcal{C}$ for the unstable scalar system that achieves $E[|X(t)|^\eta] < K$ for all bounded driving noise $-\frac{\Omega}{2} \leq W(t) \leq \frac{\Omega}{2}$, then $C_{\text{anytime}}(\eta \log_2 \lambda) \geq \log_2 \lambda$ bits per channel use for the noisy channel considered with noiseless feedback.

And for the non-nested information pattern case:

Theorem 2.4 [6] It is possible to control an unstable scalar process driven by a bounded disturbance over a noisy finite output-alphabet channel so that the η-moment of $X(t)$ stays finite for all time if the channel with feedback has $C_{\text{anytime}}(\alpha) \geq \log_2 \lambda$ for some $\alpha > \eta \log_2 \lambda$ if the observer is allowed to observe the state $X(t)$ exactly.

The encoders and decoders used in the proofs of these theorems were far from simple. Next we define the class of simplified encoders and decoders we wish to restrict ourselves to.

---

[2]Just start time at -1 if you want an unknown but bounded initial condition.

[3][7] shows how to map the results given here to almost-sure stabilization in the undisturbed case when $W(t) = 0$.

2.4 Nearly memoryless encoders and ML decoders

By nearly memoryless observers, we mean $\mathcal{O}$ functions that sample the plant state every T time units (for some $T > 0$), and then apply a channel input that depends only on the last such sample.

Definition 2.5 A *random nearly memoryless* observer is a sequence of maps $\mathcal{O}_t$ such that there exist $\mathcal{O}_t'$ so that:

$$\mathcal{O}_t(X_0^t, Z_0^t) = \mathcal{O}_t'(X(T\lfloor \frac{t}{T} \rfloor), Z_{\lfloor \frac{t}{T} \rfloor})$$

where the Z_i are the iid continuous uniform random variables on $[0, 1]$ that represent the common-randomness available at both the observer/encoder and the controller/decoder.

Definition 2.6 A *sequential decoder* for a trellis code is one that searches paths through the trellis in a sequential and causal manner.

The following result will guide us in our approach and can be found in [7].

Theorem 2.7 We can η-stabilize an unstable scalar process driven by a bounded disturbance over a discrete memoryless channel if the channel without feedback has random block-coding error exponent $E_r(R) > \eta \log_2 \lambda$ for some $R > \log_2 \lambda$ and the observer is only allowed access to the plant state where $E_r(R)$ is Gallager's random coding error exponent. Furthermore, this can be achieved with a nearly memoryless random encoder/observer.

The scheme we use in the next section will adapt from the proof of this theorem, using exactly the same encoder. However, the prior proof required ML decoding over a growing trellis at the controller. It does not take long for this scheme to become impractical due to the exponential growth in required computation. This provides the motivation for us to consider using a sequential decoder at the controller.

3 Main Result

We first state and prove the main theorem for η-th moment stabilization, and then briefly analyze the computation workload of the decoder.

3.1 Theorem and proof

Theorem 3.1 We can η-stabilize an unstable scalar process driven by a bounded disturbance over a discrete memoryless channel if the channel without feedback has random block-coding error exponent $E_r(R) > \eta \log_2 \lambda$ for some $R > \log_2 \lambda$. Furthermore, this can be achieved with the same nearly memoryless encoder/observer used in [7] and a stack-based sequential decoder/controller.

Proof: Pick a rate $R > \log_2 \lambda$ for which $E_r(R) > \eta \log_2 \lambda$. Pick T, Δ large enough so that:

$$TR > \log_2(\lambda^T + \frac{\sum_{i=0}^{\infty} \lambda^{T-i} \Omega}{\Delta})$$

This means that if $X(t)$ is known to be within any given box of size Δ, then with no controls applied, the plant state $X(t + T)$ can be in no more than 2^{TR} adjacent boxes each of size Δ. It is clear that such a T, Δ pair always exists as long as $R > \log_2 \lambda$.

Our quantizer Q will look at the plant state X with a coarseness of Δ. It is clear that we satisfy the following:

a. Knowing that $X(t)$ is in a given bin of width Δ and assuming that no controls are applied, there are at most 2^{TR} possible bins which could contain $X(t + T)$.

b. The descendants of a given bin dT time units later are all in contiguous bins and furthermore, there exists a constant K such that the total length covered by these bins is $\leq K\lambda^{dT}$.

Properties [a] and [b] above easily extend to the case when the control sequence between time $X(t)$ and $X(t+T)$ is known exactly since linearity tells us that the impact of these controls is just a translation of all the bins by a known amount. Thus, we have:

c. Conditioned on actual past controls applied, the set of possible paths that the states $X(0), X(T), X(2T), \ldots$ could have taken through the quantization bins is a subset of a trellis that has a maximum branching factor of 2^{TR}. Furthermore, the total length covered by the d-stage descendants of any particular bin is bounded above by $K\lambda^{dT}$.

Not all such paths through the trellis are necessarily possible, but all possible paths do lie within the trellis. The observer/encoder independently assigns symbols from $\mathcal{A}^T$ as time-varying labels to the bins. The probability measure used to draw the symbols should be the $E_r(R)$ achieving distribution. Thus, the labels on each bin are iid through both time and across bins.

Call two paths through the trellis disjoint with depth d if their last common node was at depth d and the paths are disjoint after that. We immediately observe:

d. If two paths are disjoint in the trellis at a depth of d, then the channel inputs corresponding to the past dT channel uses are independent of each other.

In decoding, the controller creates its trellis with the knowledge of past control inputs. Then, it decodes the channel outputs at times which are multiples of T using a version of the Stack or Zigangirov-Jelinek Algorithm that is referred to as 'Algorithm A' in [1]. This version is simple and does not consider merging of paths on the trellis.

It has been shown that if x is a channel input symbol and y is a channel output symbol, the optimal metric associated with this symbol pair is of the form $m(x, y) = \log_2 \frac{p(y|x)}{p(y)} - G$, where G is a real parameter of the algorithm. For a branch with T symbols, the metric of the branch would then be $\sum_{i=0}^{T-1} \log_2 \frac{p(y_i|x_i)}{p(y_i)} - G$. The parameter G will be used to tradeoff complexity and performance of the controller. Fix a time t and consider an error event at depth d. This represents the case that the maximum likelihood path last intersected with the true path dT time steps ago. By property [c] above, our control will be based on a state estimate that can be at most $K\lambda^{dT}$ away from the true state. Thus, we have:

e. If an error event at depth d occurs at time t, the state $|X(t + T)|$ is smaller than $K'\lambda^{(d+1)T}$ for some constant K' that does not depend on d or t.

By property [c], there are no more than 2^{dTR} possible disjoint false paths that last intersected the true path d stages ago. By the memorylessness of the channel, the log-likelihood of each path is the sum of the likelihood of the "prefix" of the path leading up to d stages ago and the "suffix" of the path from that point onward. Property [d] tells us that the channel inputs corresponding to the false paths are pairwise independent of the true inputs for the past dT channel uses. For a path that is disjoint from the true path at a depth of d to beat all paths that end up at the true final state, the false path

must have a final log-likelihood that at least beats the metric of true path at the point of divergence or beyond, until the current time. With this observation and some techniques developed by Jelinek extending the random coding error exponent arguments, we have:

Theorem 3.2 ([10]) For a decoder using the Stack Algorithm with bias parameter G, let E_d denote the event that some false path that diverged with the true path d branches ago is declared the decoded path. Then, if G satisfies the condition below for ρ such that $R = E_0'(\rho)$ and $E_0(\rho)$ is Gallager's function, $P(E_d) < L \exp_2(-dTE_r(R))$.

$$G > \frac{1+\rho}{\rho}\left[E_0(\rho) + f(\rho)\right] \text{ where } f(\rho) \triangleq \log_2 \sum_y p(y) \left[\sum_x \left[\frac{p(y|x)}{p(y)}\right]^{\frac{1}{1+\rho}} p(x)\right]^\rho$$

This result is the reason for why a sequential decoder can stabilize the same moments as an ML decoder. Now we finish proving stability of the η-th moment by applying a union bound over error depths, using the stated theorem, and observation [e].

$$
\begin{aligned}
E[|X(t+T)|^\eta] &= \sum_{d=0}^{\lfloor t/T \rfloor} P(E_d) \cdot E[|X(t+T)|^\eta | E_d] \\
&< \sum_{d=0}^{\infty} L2^{-dTE_r(R)} \cdot \left(K\lambda^{(d+1)T}\right)^\eta \\
&= LK^\eta \lambda^{T\eta} \sum_{d=0}^{\infty} 2^{-dT(E_r(R)-\eta \log_2 \lambda)} \\
&= K' < \infty
\end{aligned}
$$

The final geometric sum converges because we assume $E_r(R) > \eta \log_2 \lambda$. $\qquad\square$
For sequential decoding, the decoder has to perform a random amount of computation at each time step. The random variable of computation is of practical interest in this scenario and we explore it next.

3.2 The random variable of computation

To explain the classical view of computation, consider an infinite trellis being partitioned by the true path through the trellis and a sequential decoder running for eternity to decode the true path.

Definition 3.3 The ith *incorrect subtree* is defined by the set of paths that have their last common node with the true path at depth i into the trellis. The *random variable of computation* N_i is defined to be the number of nodes in the ith incorrect subtree that are ever visited by the sequential decoder.

Theorem 3.4 ([1], [10]) If $G = R$ and ρ is defined parametrically by the equation $R = E_0(\rho)/\rho$, then the γ-th moment of N_i is finite for $\gamma < \rho$. In particular, if $R < E_0(1)$, the expected mean of computation is finite.

In the context of stabilization, however, one would like to say something about the moments of the number of computations performed at any given time.

Definition 3.5 The random variable $C(k)$ is defined to be the number of nodes visited by the sequential decoder at time k. The r.v. $C_i(k)$ is the number of nodes in the ith incorrect subtree visited at time k.

We note the following relation between these two notions of computation. Suppose $E[N_i] = K < \infty$. Then, we have

$$\lim_{k \to \infty} E[C(k)] = \lim_{k \to \infty} E[\sum_{i=1}^{k-1} C_i(k)]$$

$$\leq \lim_{k \to \infty} \sum_{n=0}^{k-2} \sup_{l>0} E[C_l(n+l)]$$

There are more partial paths on the stack as time increases, and the metrics of the incorrect subtree relative to the correct path are the same regardless of i, so the expected value of $C_l(n+l)$ should be insensitive to l since the averaging is performed over all possible random codes and channel noise. Thus, we have

$$\lim_{k \to \infty} E[C(k)] \leq \sum_{n=0}^{\infty} E[C_1(n+1)]$$

$$= E[\sum_{n=0}^{\infty} C_1(n+1)]]$$

$$= E[N_1] < \infty$$

Having shown that the mean of $C(k)$ stays bounded, we conjecture that if a moment of N_i exists for all i, the same moment of $C(k)$ will exist for all k.
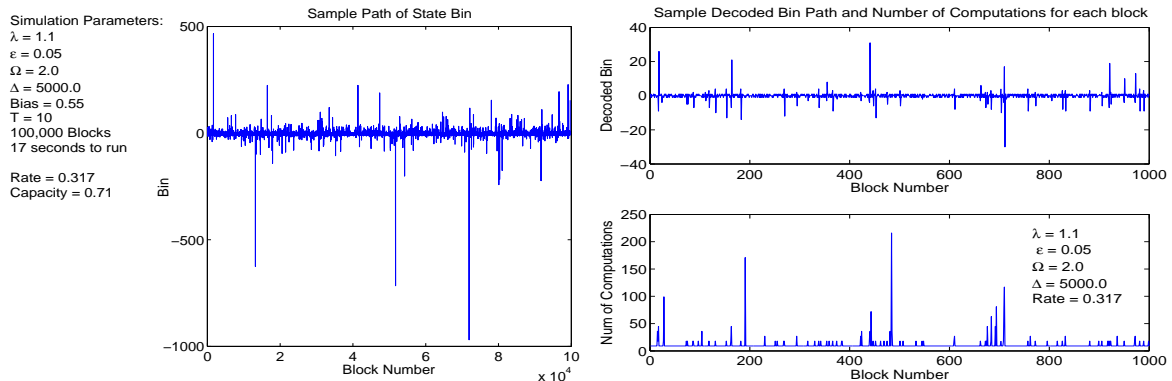

Figure 2: Left: A sample path of the actual bin the state lies in over 100,000 branches of length 10. Right: A sample path of the decoded bin and C(k). Note that large deviations in the decoded bin from 0 generally occur near high periods of computation.

4   Simulation Results

The main advantage of using sequential decoding over ML decoding, as we have claimed, is that the complexity of the decoder does not grow unboundedly as time goes on, except for the requirement of memory that grows roughly linearly with time. By simulating the strategy described in Theorem 3.1, we show that complexity is reduced to such an

extent as to make the scheme practical. The simulations were performed with a BSC with crossover probability ϵ as the channel. Also, we note that the rate is defined to be $\log_2(B)/T$ where B is the branching factor of the decoding trellis over a block of T symbols.

4.1 Sample paths

Since the overall system 'operates' at a resolution of Δ, it is not interesting to look at both the state and the bin of the state, and so we focus on the actual and decoded bins, rather than the state. Figure 2 shows a sample path of the actual bin, and a sample path of the decoded bin along with a sample path of $C(k)$. The sample path shows that large deviations of the state from 0 are rare. By applying Markov's inequality using a moment η that we know can be stabilized, we have

$$P(|X(t)|^\eta \geq x^\eta) \leq K_s x^{-\eta}$$

Similarly, $P(C(k) \geq N) \leq K_c N^{-\gamma}$ if we know the γ-th moment of computation to be finite. Viewed on a log-log plot, both these quantities should decay linearly with a slope equal to the supremum of finite moments. In Figure 3, we see that with a bias that is optimized to achieve the random coding exponent for probability of error, the computation[4] does not have the moment stability guaranteed by Theorem 3.4 if we had the bias equal to the rate. Also, the power law for the state is significantly better than guaranteed by the random coding exponent, which is $E_r(R)/\log_2(\lambda)$.
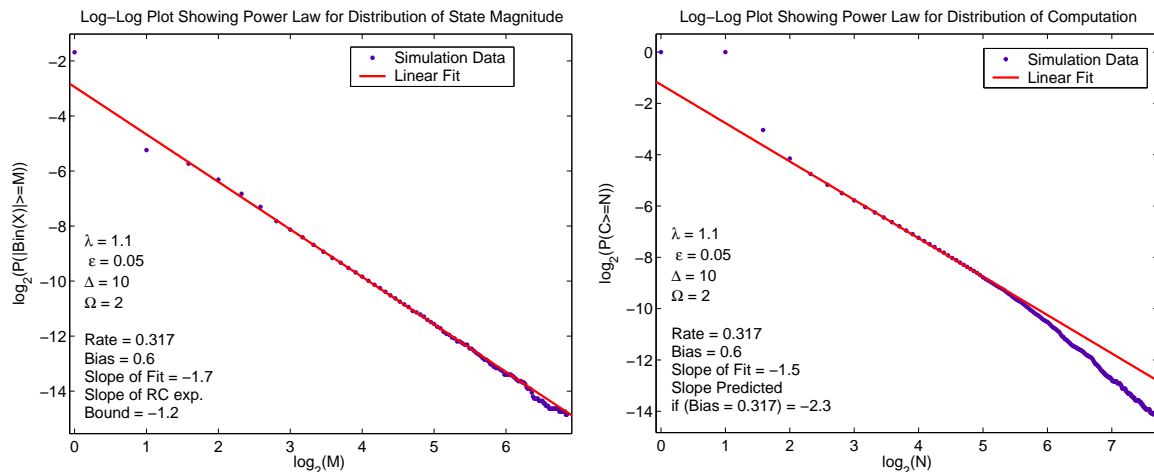

Figure 3: Left: Log-log plot of the complementary cdf of the state. Right: Log-log plot of the random variable of computation.

4.2 Tradeoff between computation and stability

As noted in [10], it is *not possible* in general to optimize both probability of depth d error and computation simultaneously with a single choice of bias. For the BSC, the bias required to achieve the random coding exponent is always greater than the rate of the trellis code.

---

[4]In all plots, computation refers to $C(k)$. We have made the assumption of ergodicity as well as that the moments of the two computations are equal asymptotically. Although this hasn't been proven, it is plausible because the simulations show that $C(k)$ is not growing unbounded over time.

Figure 4 shows the tradeoff between these two metrics of performance as a function of the bias parameter. As expected, increasing the bias always improves state stability. However, decreasing the bias does not always improve the moments of computation. By decreasing the bias below the rate, we run the risk of forging ahead into the trellis along paths that are not correct, only to have to return to the nodes we passed over later on.
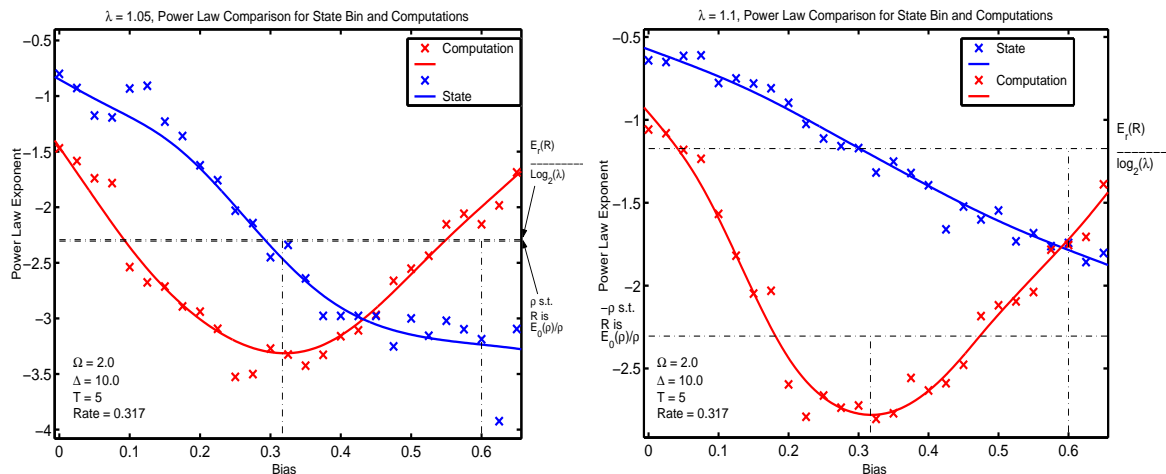

Figure 4: Left: Plot of the computation and state stability tradeoff for $\lambda = 1.05$. Right: Plot of the computation and state stability tradeoff for $\lambda = 1.1$.

4.3 Comments

An implicit assumption made is that the controller/decoder has unlimited computational speed so that no matter how many computations must be performed at one time, the decoder can finish them and apply the control to the plant before the next time step. Suppose instead that no more than $L - 1$ computations may be performed within any T time steps. If there are more than NL computations required at one time, the controller takes no action for N blocks, leaving the state to wander until it can decode. It is known that the distribution of computation can be lower bounded by a Pareto distribution[4] with some parameter γ. So even in the case that the noise is negligible, assuming $|X(t)| = x$, we have

$$
E\left[|X(t+NT)| \Big| |X(t)| = x\right] \geq E\left[|X(t+NT)| \Big| |X(t)| = x, C(t) > NL\right] P(C(t) > NL)
$$
$$
\geq xK_c \cdot \lambda^{NT} N^{-\gamma}
$$

Letting N go to infinity, we see that expected value of the state diverges. Clearly, this applies to any moment of the state. So this scheme fails to stabilize the state if the controller has limited computational speed, regardless of how large that limit. However, for the binary erasure channel, one *can* use a finite speed decoder and essentially memoryless encoder to control the state.[5] This leads us to wonder whether there is a way stabilize with finite speed decoding for arbitrary DMCs.

5   Conclusion

We have developed a strategy for stabilization over a noisy channel that works in a low complexity way at both the encoder and decoder. We have shown that a sequential

decoder can be used in place of an ML decoder with almost the same performance in stabilizing the system. Furthermore, the amount of computation performed by the decoder fits with the concept of a stable system, namely that we shouldn't have to do an ever increasing amount of work if the system keeps returning to the same few states.

There are many more questions that can be naturally asked about this problem. The most practically important of these is the question of performance. In the low rate regime, is there a way of using some bits to supervise the state into remaining near the center of a coarser bin? Such a controller would essentially operate at different time scales, decoupling the questions of performance and asymptotic stability. Second, and perhaps of greater theoretical interest, if we don't restrict ourselves to a uniform quantizer, could we get away with using less rate or stabilizing higher moments? It turns out that there is a large gap between the performance being achieved by this scheme and the bound given by the reliability exponent for a channel with feedback.[8]

References

[1] G.D. Forney, Jr., "Convolutional codes III. Sequential decoding." *Inform. and Control*, vol. 25 pp. 267-297, 1974.

[2] R.G. Gallager, *Information Theory and Reliable Communication*. New York, NY: John Wiley and Sons, 1971.

[3] Y.C. Ho, M.P. Kastner, and E. Wong, "Teams, Signaling, and Information Theory," *IEEE Transactions on Automatic Control*, April 1978, pp 305-312.

[4] I.M. Jacobs and E.R. Berlekamp, "A lower bound to the distribution of computation for sequential decoding," *IEEE Transactions on Information Theory*, vol. 13, pp. 167-174, April 1967.

[5] A. Sahai, "Evaluating channels for control: Capacity reconsidered," in *Proceedings of the 2000 American Control Conference*, Chicago, CA, June 2000, pp. 2358–2362.

[6] A. Sahai, "The necessity and sufficiency of anytime capacity for control over a noisy communication link," *Proceedings of the 43rd IEEE Conference on Decision and Control*, December 2004.

[7] A. Sahai and S.K. Mitter, "The necessity and sufficiency of anytime capacity for control over a noisy communication link: Part I" Submitted to the *IEEE Transactions on Information Theory*, April 2005.

[8] A. Sahai, T. Simsek, and P. Varaiya, "Why block length and delay are not the same thing." In preparation.

[9] H. Witsenhausen, "Separation of Estimation and Control for Discrete Time Systems." *Proceedings of the IEEE*, Vol 59, No. 11, November 1971.

[10] F. Jelinek "Upper Bounds on Sequential Decoding Performance Parameters," *IEEE Transactions on Information Theory*, vol. 20, pp. 227-239, Mar. 1974.