Towards the Control of Linear Systems with Minimum Bit-Rate

João Hespanha[†] hespanha@ece.ucsb.edu

Antonio Ortega[‡] ortega@sipi.usc.edu

Lavanya Vasudevan[‡] vasudeva@usc.edu

[†]*Dept. Electrical & Computer Engineering, Univ. of California, Santa Barbara, CA 93106-9560*

[‡]*Dept. Electrical Engineering, Univ. of Southern California, Los Angeles, CA 90089*

Abstract

We address the problem of determining the minimum bit-rate needed to stabilize a linear time-invariant process. For the noise free case, we determine a bit-rate below which stabilization is not possible and above which asymptotic stabilization can be achieved. Inspired by differential pulse code modulation (DPCM) techniques, we propose practical encoding/decoding schemes that guarantee boundedness of the state for the case of a noisy linear time-invariant process. With fixed-step quantization, we are only able to approach the minimum bit-rate for the noiseless case. However, with variable-step quantization we are able to approach it even in the presence of noise and disturbances.

1 Introduction

In this paper we address the problem of stabilizing a linear process through feedback under constrains on the bit-rate of the feedback loop. We assume that the actuator and the sensor used to measure the process' output are connected though a communication channel with finite bandwidth, measured in terms of the maximum number of bits per second (bit-rate) that it can carry. In this context we pose several questions; answering them will lead us to an efficient coding design:

1. *What is the minimum bit-rate for which stabilization is possible?*
2. *In cases where the system state is multidimensional, should a different number of bits be used to encode different dimensions of the state? if so, what are the right bit allocations?*
3. *Should the quantization step size be fixed or should it vary according to the state of the system?*

We address these issues for linear time-invariant processes in the presence of (deterministic) input disturbances and measurement noise. For simplicity, we assume that the sensors provide (noisy) state measurements but most of the results would generalize for the case of output-feedback through the use of state-estimators co-located with the sensors. The problem of stabilization with finite communication bandwidth was introduced by Wong and Brockett [9, 10] and further pursued by [6, 8, 7, 4]. The main novelty of our work is the introduction and analysis of practical quantization schemes for communication-constrained stabilization under noisy conditions.

In Section 2, we address the question of determining the minimum bit-rate for which stabilization of a noiseless process is possible. We compute a minimum average bit-rate below which it is not possible to stabilize the process. This rate is determined by the unstable eigenvalues of the open-loop process. We also show that this lower-bound for the bit-rate is tight in the sense that it is

[†]This material is based upon work supported by the National Science Foundation under Grant No. ECS-0093762.

possible to achieve asymptotic stability for any rate above it. In defining bit-rate, our analysis in Section 2 considers an approach where one bit is used for each sampling time. However we do not restrict our attention to transmitting single bits uniformly distributed over time. In fact, our negative results state that stabilization is not possible if the *average* bit-rate falls below a certain threshold $f_{\min}$. This means that, e.g., using an N-symbol alphabet and sending k symbols every time interval of length T, stabilization is only possible when $k \log_2 N/T \geq f_{\min}$. It is important to clarify that we are equating stability with boundedness and therefore, when we say that stabilization is not possible, we actually mean that the state will grow unbounded for some initial conditions.

In Section 3, we consider the design of encoder/decoder pairs for linear processes whose state can be measured by noisy sensors and with an additive input disturbance that cannot be measured. The approach pursued is inspired by the derivations in Section 2, where bit-rate efficient coding is based on encoder/decoder pairs that have an internal state that is kept synchronized. This intuition leads to the design of a novel predictive coding scheme, similar to standard Differential Pulse Code Modulation (DPCM) [3], where the predictor incorporates all available knowledge of the dynamics of the system. In our case, since the goal is for both encoder and decoder to track the state of the system, our predictor is constructed using the reconstructed state data to estimate the expected state of the system at the next sampling time. Then the encoder will transmit to the decoder the quantized difference between actual state and predicted state. By quantizing an estimation error instead of the process' state, we are able to significantly reduce the bit-rate needed to stabilize the process (cf., e.g., [2, 4]).

We propose two distinct predictive encoding/decoding schemes and study their performances. The first one (cf. Section 3.1) employs fixed-step quantization, which means that the number of, and distance between, quantization steps remain constant. This scheme is simpler but can lead to poor performance, especially for large initial conditions of the process. To overcome this difficulty we also propose a encoding/decoding scheme with variable-step quantization, in which the number of quantization steps remains constant but the distance between steps may vary (cf. Section 3.2). This allows us to improve the quality of the steady-state response of the closed-loop system, with respect to a fixed quantization step-size at the same coding rate.

Section 4 presents two example systems to illustrate the efficiency of our work. One of the examples is based on the predictive coding introduced in Section 3 and is applied to a one-dimensional process. The second example deals with a two-dimensional process, where we make use of the analysis to demonstrate the importance of bit allocation (i.e., using different number of bits for each state dimension, in accordance with its importance for system stability.)

Related work

The problem of determining the minimum bit-rate to achieve stability has been addressed in [6, 8]. Nair and Evans [6] considered the stabilization of infinite-dimensional discrete-time linear processes. They considered the noiseless case, where the initial condition is a scalar random variable. When transposed to the finite-dimensional case, the minimum-rate found in [6] is actually smaller than the one determined here. However, this is an artifact of considering a single scalar initial condition. Tatikonda [8] considered the stabilization of finite-dimensional discrete-time noiseless linear processes. We provide analogous results for continuous-time systems. The formulas derived here match the ones that would have been obtained from [8] for the discrete-time process obtained

from sampling the continuous-time one. This shows that sampling does not have a negative impact on the minimum bit-rate needed for stabilization (assuming that controllability is not lost through sampling). Our negative results on non-stabilizability are somewhat stronger than those in [8] because we consider more general classes of encoders (e.g., not necessarily a fixed number of bits per sampling interval) and a less restrictive notion stability (just boundedness as opposed to Lyapunov asymptotic stability).

The use of a prediction-based encoder was also proposed by Liberzon [4], where a ball around the most updated state-estimate is broken into square boxes (one for each symbol in the alphabet). However, since there is no attempt to do optimal relative bit allocation, the bit-rates required by [4] are higher than the ones needed here when considering states of dimension higher than one. This penalty is paid due to using the same number of bits to represent all dimensions of the state instead of, as we propose, using for each dimension the minimum number of bits that will guarantee convergence.

The use of variable-step quantizers was inspired by the work of Brockett and Liberzon [1], where they are used to achieve global asymptotic stability. As we do here, they decrease the step-size as the system approaches the origin to achieve convergence to zero. To achieve global results, they increase the step-size when the system's state is outside the range of the quantizer. This technique could also be used here to make our results global. In [1] no attempt is made to reduce the bit-rate, as the goal of that work is to study the stabilization of quantized systems. By using variable-step quantizers, we were able to stabilize linear processes affected by additive disturbances and measurement noise, at any rate above the minimal one that was determined for the noiseless case. This shows that additive noise/disturbances do not have a negative impact on the minimum bit-rate needed for stabilization.

2 Minimum bit-rate

Consider a stabilizable linear time-invariant process

$$\dot{x} = Ax + Bu, \qquad x \in \mathbb{R}^n, u \in \mathbb{R}^m, \tag{2.1}$$

for which it is known that $x(0)$ belongs to some bounded measurable set $\mathcal{X}_0 \subset \mathbb{R}^n$ with positive measure. We are interested in stabilizing this process using a control law that utilizes limited information about the state. In particular, we assume that we have an encoder that observes $x(t)$ and generates a stream of bits $\{b_k : k = 0, 1, 2, \dots\}$ at times $\{t_k : k = 0, 1, 2, \dots\}$. The control signal $u(t)$ is then determined solely from observing the stream $\{b_k\}$ by a pair decoder/controller (cf. Figure 1). One can then ask the question: *What is the minimum bit-rate for which one can stabilize the closed-loop system?*

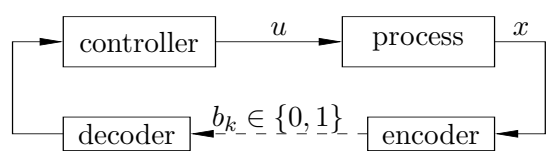

Figure 1: Closed-loop with binary feedback path

Clearly, when A is stable no information is needed to stabilize (2.1) as one can just set $u(t) = 0$, $t \geq 0$. Also, when A has a stable A-invariant subspace, one can assume that $x(0)$ has no component on that subspace because even if this is not the case, such component will decay exponentially fast to zero. We will therefore consider the most interesting case in which all the eigenvalues of A have nonnegative real part.

Let us put ourselves in the position of the decoder/controller that observes the sequence of bits $\{b_k\}$ and attempts to generate u. In case x was precisely known at some time t_k, it would be possible to compute an open loop law $u(t)$, $t \geq t_k$ that would drive x to zero (even in finite time). However, this is not possible because the k bits available up to time t_k are not sufficient to determine the precise value of $x(t_k)$. In practice, with k bits one is only able to determine that $x(t_k)$ belongs to some set $\mathcal{X}_k \subset \mathbb{R}^n$. If the sequence of sets $\{\mathcal{X}_k\}$ grows as $k \to \infty$, it is impossible to stabilize the system because, in the worst case, one could always be in a point of the set that is at an increasing distance away from the origin. However, if the sequence $\{\mathcal{X}_k\}$ is bounded then it is possible to stabilize the system because, by choosing appropriate controls, one can steer the system so that the $\mathcal{X}_k$ are always centered at the origin and therefore the state remains bounded. So the question formulated above is equivalent to: *What is the minimum bit-rate for which one can keep the sequence of sets $\{\mathcal{X}_k\}$ uniformly bounded?*

We start by searching for a negative result, i.e., a bit-rate below which the sequence of sets is unbounded. Actually, it is slightly easier to search for a bit-rate below which the "volume" of the sets $\{\mathcal{X}_k\}$ is unbounded. To this effect, consider two instants of time t_k, t_{k+1} at which consecutive bits were received[1] and let $\mu(\mathcal{X}_k) := \int_{x \in \mathcal{X}_k} dx$. be the volume of the set to which $x(t_k)$ is known to belong immediately after the bit b_k is received. By the variation of constants formula

$$x(t_{k+1}) = e^{A(t_{k+1}-t_k)} x(t_k) + u_k,$$

where $u_k := \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} Bu(\tau) d\tau$. Therefore, before the bit b_{k+1} is received, $x(t_{k+1})$ is only known to be in the set $\mathcal{X}_{k+1}^- := e^{A(t_{k+1}-t_k)} \mathcal{X}_k + u_k$. We are assuming here that the decoder knows the input u that it applied during this period. The volume of $\mathcal{X}_{k+1}^-$ is then given by

$$\mu(\mathcal{X}_{k+1}^-) := \int_{x \in e^{A(t_{k+1}-t_k)} \mathcal{X}_k + u_k} dx = \int_{z \in \mathcal{X}_k} |\det e^{A(t_{k+1}-t_k)}| dz = |\det e^{A(t_{k+1}-t_k)}| \mu(\mathcal{X}_k).$$

Here, we made the change of integration variable $z := e^{A(t_{k+1}-t_k)}(x - u_k)$. Using the fact that $\det e^M = e^{\operatorname{trace} M} = e^{\sum_{i=1}^n \lambda_i[M]}$ for any $n \times n$ matrix M with eigenvalues $\lambda_1[M], \lambda_2[M], \ldots, \lambda_n[M]$, we conclude that[2]

$$\mu(\mathcal{X}_{k+1}^-) = e^{(t_{k+1}-t_k) \operatorname{trace} A} \mu(\mathcal{X}_k) = e^{(t_{k+1}-t_k) \sum_{i=1}^n \lambda_i[A]} \mu(\mathcal{X}_k).$$

Suppose now that the coding is optimal in the specific sense that it would minimize the volume of the set $\mathcal{X}_{k+1}$ to which $x(t_{k+1})$ will be known to belong. To maximize the volume reduction provided

[1] The times t_k and t_{k+1} need not be distinct. In fact, we should regard using an alphabet with $N > 2$ symbols, as receiving several bits simultaneously.

[2] In case we were working on discrete-time, we would have

$$\mu(\mathcal{X}_{k+1}^-) = |\det A^{t_{k+1}-t_k}| \, \mu(\mathcal{X}_k) = |\det A|^{t_{k+1}-t_k} \, \mu(\mathcal{X}_k) = \Big| \prod_{i=1}^n \lambda_i[A] \Big|^{t_{k+1}-t_k} \mu(\mathcal{X}_k).$$

by bit b_{k+1}, this bit should allow us to locate $x(t_{k+1})$ in one of two half-volumes of $\mathcal{X}_{k+1}^-$. Thus,

$$\mu(\mathcal{X}_{k+1}) \geq \frac{1}{2}\mu(\mathcal{X}_{k+1}^-) = \frac{1}{2}e^{(t_{k+1}-t_k)\sum_{i=1}^n \lambda_i[A]}\,\mu(\mathcal{X}_k),$$

with equality achievable though *optimal volume-reduction coding*. Iterating this formula from 0 to k, we conclude that

$$\mu(\mathcal{X}_k) \geq e^{t_k\left(\sum_{i=1}^n \lambda_i[A]\right)-k\log 2}\,\mu(\mathcal{X}_0).$$

Suppose then that

$$\lim_{k\to\infty} t_k\left(\sum_{i=1}^n \lambda_i[A]\right) - k\log 2 = +\infty. \tag{2.2}$$

This would mean that the volume of the sets $\{\mathcal{X}_k\}$ grows to infinity as $k \to \infty$, which in turn means that we can find points in these sets arbitrarily far apart for sufficiently large k. In this case, stability is not possible because the state could be arbitrarily far from the origin regardless of the control signal used. Moreover, any other encoding would also necessarily lead to unbounded volumes and therefore instability because it could do no better than the optimal volume-reduction coding considered. To avoid (2.2) – which would necessarily lead to instability – the sequence t_k must grow no faster than $k\log 2/\sum_{i=1}^n \lambda_i[A]$ for stability to be possible. This leads to the following result:

Theorem 1. *It is not possible to stabilize the process* (2.1) *with an average bit-rate smaller than*

$$r_{\min} := \frac{1}{\log 2}\sum_{i:\Re\lambda_i[A]\geq 0} \lambda_i[A].$$

For a discrete-time process $r_{\min}$ *is given by*

$$r_{\min} := \frac{1}{\log 2}\log\Big|\prod_{i:|\lambda_i[A]|\geq 1} \lambda_i[A]\Big|.$$

We show in Section 3 how to apply some of the ideas derived from this ideal case to the design a practical coding system.

Optimal Encoding

Suppose now that the average bit-rate is larger than $r_{\min}$ and therefore that

$$\lim_{k\to\infty} t_k\left(\sum_{i=1}^n \lambda_i[A]\right) - k\log 2 = -\infty,$$

which means that the volume of the sequence of sets $\{\mathcal{X}_k\}$ converges to zero. It turns out that this is not sufficient to guarantee that these sets are uniformly bounded. In fact, their diameter could be increasing if the sets were becoming increasingly "thin." We show next that this does not happen when one chooses an appropriate encoding scheme. For simplicity we assume that the matrix A

is diagonal[3] and use the norm $\|x\|_\infty = \max_i |x_i|$ to compute the diameter of sets. Suppose that one utilizes an encoding scheme that first places a bounding hyper-rectangle $\mathcal{R}_k^-$ around $\mathcal{X}_k^-$, then partitions this hyper-rectangle in two sub-rectangles $\mathcal{R}_k^0$ and $\mathcal{R}_k^1$ along the largest axis of $\mathcal{R}_k^-$, and finally selects

$$b_k = \begin{cases} 0 & x(t_k) \in \mathcal{R}_k^0 \\ 1 & x(t_k) \in \mathcal{R}_k^1 \end{cases}.$$

Using this coding scheme, $\mathcal{X}_k$ is necessarily inside the hyper-rectangle $\mathcal{R}_k := \mathcal{R}_k^{b_k}$. We show next that all the axes of these hyper-rectangles converge to zero as $k \to \infty$. To this effect, for each $i \in \{1, 2, \ldots, n\}$ let a_k^i denote the size of the ith axis of the hyper-rectangle $\mathcal{R}_k$ and $r^i \in [0, 1]$ the average rate at which the ith axis is divided by two in the encoding scheme (i.e., the rate at which this axis is the largest). These quantities can be related by

$$\lim_{k\to\infty} \frac{e^{\lambda_i t_k}}{2^{r^i k}} a_0^i - a_k^i = \lim_{k\to\infty} e^{\lambda_i t_k - r^i k \log 2} a_0^i - a_k^i = 0, \qquad i \in \{1, 2, \ldots, n\}.$$

By contradiction suppose that m of the a_k^i are not converging to zero (the first m for simplicity) and that the remaining are. This means that, for each $i \in \{1, 2, \ldots, m\}$,

$$\limsup_{k\to\infty} e^{\lambda_i t_k - r^i k \log 2} > 0 \quad \Leftrightarrow \quad \limsup_{k\to\infty} \lambda_i t_k - r^i k \log 2 > -\infty \quad \Rightarrow \quad r^i \leq \frac{\lambda_i}{\log 2} \limsup_{k\to\infty} \frac{t_k}{k}. \tag{2.3}$$

Moreover, $\sum_{i=1}^m r^i = 1$ because these must be the largest axes after some finite time. Therefore, we must have

$$1 \leq \frac{1}{\log 2} \Big(\sum_{i=1}^m \lambda_i \Big) \limsup_{k\to\infty} \frac{t_k}{k}.$$

Since the axes that are not converging to zero they must correspond to unstable eigenvectors, and we conclude that

$$r_{\min} := \frac{1}{\log 2} \sum_{i:\Re\lambda_i[A]\geq 0} \lambda_i[A] \geq \frac{1}{\log 2} \sum_{i=1}^m \lambda_i \geq \limsup_{k\to\infty} \frac{k}{t_k}.$$

Since this contradicts the fact that the average bit-rate is larger than $r_{\min}$, we conclude that all axes must be converging to zero. Since $x(t_k)$ is known to be inside $\mathcal{R}_k$, this means that as $k \to \infty$ the decoder/controller can determine the precise value of the state and drive it to the origin. The following can then be added:

Theorem 2. *Assume that A is diagonalizable:*

1. *It is possible to asymptotically stabilize* (2.1) *with any bit-rate larger than $r_{\min}$.*
2. *It is possible to stabilize*[4] (2.1) *with a bit-rate equal to $r_{\min}$.*

[3] If this was not the case but A was diagonalizable we could always consider a different norm that is obtained from $\|x\|_\infty$ through a similarity transformation. In case A had complex conjugate eigenvalues, the pair of conjugate eigenvalues would have to be treated jointly but the results would still hold (cf. Remark 7 in Section 3.1).

[4] To prove this, one would replace the three inequalities in (2.3) by $\cdot = +\infty$, $\cdot = +\infty$, and $\cdot < \cdot$, also arriving at a contradiction.

Remark 3. Diagonalizability is almost certainly not needed for part 1 and almost certainly needed for 2. Uniformity in the convergence could be achieved if one required some form of uniformity in the sampling (cf. Remark 9 in Section 3.2).

It is interesting to compare the results above with those in Elia and Mitter [2], where it is shown that the memoryless quantization scheme that minimizes the product of quantization density times sampling rate requires a sampling rate of

$$\frac{1}{T^*} := \frac{1}{\log(1+\sqrt{2})} \sum_{i:\Re\lambda_i[A]\geq 0} \lambda_i[A] \approx \frac{r_{\min}}{1.27}.$$

Theorem 2 indicates that, on average, only 1.27 bits need to be transmitted at this rate, when one considers encoding/decoding schemes with memory. Liberzon [4] also proposes an encoding scheme for stabilization under limited communication. For a diagonal (or diagonalizable) matrix A, that scheme would need an average bit-rate strictly larger than $\frac{n\,\Re\lambda_{\max}[A]}{\log 2}$, where $\lambda_{\max}[A]$ denotes the (unstable) eigenvalue of A with largest real-part. For a one-dimensional process this bit-rate is precisely $r_{\min}$, but for higher dimensions it is generally larger than $r_{\min}$.

Remark 4. When the state is not directly measurable but only accessible through an output $y := Cx + Du$ (through which the system is assumed detectable), the same bounds on the minimum stabilizable bit-rates would hold because one could embed a state estimator in the encoder. As before, we assume that the encoder knows the control law that is being used and can therefore "guess" the control signal being applied.

Remark 5. A key observation to be made is that the encoder and decoder used to achieve optimality had an internal state that was kept synchronized. In the above discussion this was implicit, as we assumed that encoder and decoder both had knowledge of the rectangles $\mathcal{R}_k$ where the state was known to be at time t_k. In the next section we use this idea to introduce explicitly state prediction in the coding algorithm.

3 State-prediction coding

Consider now a linear process with measurement noise and input disturbance given by

$$\dot{x} = Ax + B(u + \mathbf{d}), \qquad y = x + \mathbf{n}, \qquad x, \mathbf{n} \in \mathbb{R}^n,\ u, \mathbf{d} \in \mathbb{R}^m, \tag{3.4}$$

where $\mathbf{d}$ denotes an exogenous disturbance and w measurement noise. Both $\mathbf{n}$ and $\mathbf{d}$ cannot be directly measured but are assumed bounded. For generality we assume that the symbols $\{w_k : k = 0, 1, 2, \dots\}$ generated by the encoder at the times $\{t_k : k = 0, 1, 2, \dots\}$ are not necessarily binary. Instead, each w_k is a symbol in a finite alphabet with N distinct symbols. For simplicity we now assume that the times t_k are equally spaced by a fixed sampling interval T_s. The corresponding feedback loop is shown in Figure 2. We restrict our attention to linear state-feedback controllers of the form

$$u = Ky, \tag{3.5}$$

where K is a state-feedback matrix that makes $A+BK$ asymptotically stable. Clearly, (3.5) cannot be implemented because y is not available to the controller. We use instead

$$u = Ky_q, \tag{3.6}$$

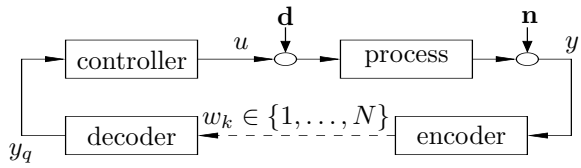

Figure 2: Closed-loop with process' state encoding

where y_q denotes the decoder's output.

Based on Remark 5, we propose a specific encoder/decoder pair to stabilize (3.4) under communication constrains. This system is similar to standard Differential Pulse Code Modulation (DPCM) techniques in that a prediction is generated by both encoder and decoder based on past quantized data. However, a key difference is that the system we consider has a deterministic component that controls the evolution of the state. Thus our predictor is meant to track the state of the system based on information available at encoder and decoder (including both past state and the various matrices that define the system dynamics.) Then, as in standard DPCM schemes, the difference between the predicted and the measured stated values is quantized at the encoder and transmitted to the decoder, thus reducing the number of bits required. Because encoder and decoder produce state predictions based on quantized data, both can generate the same prediction.

We place inside the encoder and decoder identical "copies" of the process and controller that are used to construct an estimate $\hat{x}$ of the process' state x. This can be done using the following differential equation:

$$\dot{\hat{x}} = A\hat{x} + Bu, \qquad u = K\hat{x}. \tag{3.7}$$

At each sampling time t_k, the difference $e_y := y - \hat{x}$ is quantized into one of N levels, producing $e_q := Q(e_y)$, where Q denoted a static quantization function. The corresponding word w_k is then transmitted to the decoder. Both in encoder and decoder, the state estimate $\hat{x}$ is updated by

$$\hat{x}(t_k) = \hat{x}^-(t_k) + e_q^-(t_k) = \hat{x}^-(t_k) + Q\big(e_y^-(t_k)\big) = \hat{x}^-(t_k) + Q\big(y(t_k) - \hat{x}^-(t_k)\big),$$

where the superscript $^-$ refers to the limit from below. Note that if Q was the identity (no quantization), we would have $\hat{x}(t_k) = y(t_k)$. Because of the quantization error, we actually have

$$\hat{x}(t_k) = y(t_k) - \Delta Q\big(y(t_k) - \hat{x}^-(t_k)\big),$$

where $\Delta Q(z) := z - Q(z)$, $\forall z \in \mathbb{R}^n$. The encoder and decoder just described are shown in Figure 3. We will refer to these as a *state-prediction encoder/decoder pair*. With some abuse of notation we used the same symbols to denote those signals inside the decoder and the encoder that will always remain equal (assuming that there are no digital transmission errors).

3.1 Fixed-step quantization

We consider now the problem of designing a quantizer Q for which we have boundedness of the process' state x. To this effect let us define e to be the error between the process' state x and its estimate $\hat{x}$, i.e., $e := x - \hat{x}$. From (3.4) and (3.7), we conclude that between sampling times

$$\dot{e} = Ae + B\mathbf{d}, \tag{3.8}$$

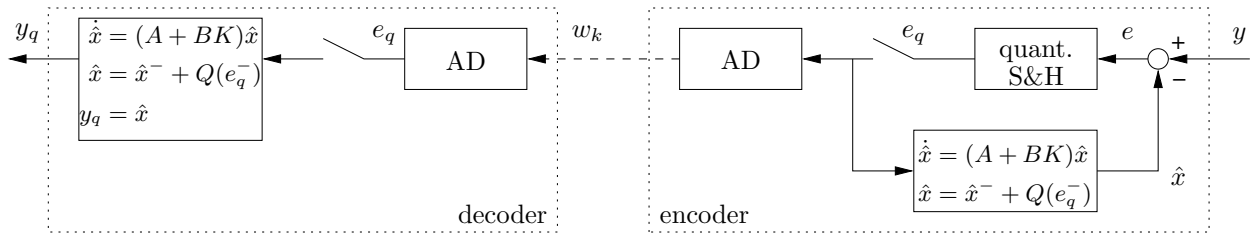

Figure 3: State-prediction encoder/decoder pair

and at a sampling time t_k, $e(t_k) = \Delta Q\big(e^-(t_k) + \mathbf{n}(t_k)\big) - \mathbf{n}(t_k)$. Denoting by T_s a fixed sampling interval and by t_k and $t_{k+1} := t_k + T_s$ consecutive sampling times, we obtain

$$e(t) = e^{A(t-t_k)}\Delta Q\big(e^-(t_k) + \mathbf{n}(t_k)\big) + \mathbf{w}(t, t_k), \qquad \forall t \in [t_k, t_{k+1}), \tag{3.9}$$

where $\mathbf{w}(t,\tau) := \int_\tau^t e^{A(t-s)}B\mathbf{d}(s)ds - e^{A(t-\tau)}\mathbf{n}(\tau)$. From (3.4) and (3.6) we also conclude that

$$\dot{x} = (A+BK)x - BKe + B\mathbf{d}, \tag{3.10}$$

which shows that the estimation error e can be viewed as additive measurement noise on a non-quantized closed-loop.

For simplicity, we assume for now that A is diagonal with real eigenvalues $\lambda_i[A]$, $i \in \{1, 2, \ldots, n\}$. The fact that A is assumed diagonal introduces no loss of generality as long as A is diagonalizable. The results that follow can also be adapted to handle the case when A has complex eigenvalues (cf. Remark 7 below).

We consider here quantizers of the form

$$Q_{K,\ell}(e) = \begin{bmatrix} \ell_1\, q_{K_1}(e_1/\ell_1) \\ \ell_2\, q_{K_2}(e_2/\ell_2) \\ \vdots \\ \ell_n\, q_{K_n}(e_n/\ell_n) \end{bmatrix}, \qquad e \in \mathbb{R}^n,\ K \in \mathbb{N}^n_{>0}, \ell \in (0,\infty)^n$$

where $q_{K_i} : \mathbb{R} \to \mathbb{R}$, $K_i \in \mathbb{N}_{>0}$ denotes a uniform quantizer of the interval $[-1, 1]$ into K_i levels. Therefore, defining $\Delta q_{K_i}(s) := s - q_{K_i}(s)$, $\forall s \in \mathbb{R}$ we have that

$$|\Delta q_{K_i}(s)| \le \frac{1}{K_i}, \qquad \forall s \in [-1, 1]. \tag{3.11}$$

The following theorem provides conditions on the quantizer that guarantee boundedness of the state of the closed-loop system. These conditions came in terms of the minimum number of quantization levels needed for each component of the state, thus implicitly defining how the bit-rate should be divided among the state components.

Theorem 6 (Bit-allocation for fixed-step quantization). *Let $w_i \ge 0$ and $n_i \ge 0$ denote upper bounds on the absolute values of the ith component of $\mathbf{w}(t,\tau)$, $\tau + T_s \ge t \ge \tau \ge 0$ and $\mathbf{n}(t)$, $t \ge 0$, respectively, and $\gamma \in (0, 1]$, $K \in \mathbb{N}^n_{>0}$, $\ell \in (0,\infty)^n$ be such that*

$$K_i \ge \frac{\ell_i\, e^{\lambda_i[A]T_s}}{\gamma(\ell_i - n_i) - w_i}, \qquad \gamma(\ell_i - n_i) \ge w_i, \qquad \forall i \in \{1, \ldots, n\}, \tag{3.12}$$

and consider the quantizer $Q := Q_{K,\ell}$. Initializing the encoder and decoder with $\hat{x}^-(t_0) = 0$, where t_0 denotes the first sampling time, and assuming that

$$|x_i(t_0)| \leq \ell_i - n_i, \qquad \forall i \in \{1, \ldots, n\}, \tag{3.13}$$

we have that

$$|e_i(t)| \leq \gamma(\ell_i - n_i), \qquad \forall t \geq t_0,\ i \in \{1, \ldots, n\}, \tag{3.14}$$

and the process' state x remains uniformly bounded.

Before proving Theorem 6, it is instructive to compare it with Theorem 1. To this effect, note that the encoder proposed here requires $N := \prod_i K_i$ symbols to be transmitted every T_s seconds. This corresponds to an average bit-rate of

$$r := \frac{\log \prod_i K_i}{T_s \log 2} = \frac{\sum_i \log K_i}{T_s \log 2}.$$

According to Theorem 6, we can choose K_i as low as $\min\{1, \frac{\ell_i\, e^{\lambda_i[A]T_s}}{\gamma(\ell_i - n_i) - w_i}\}$, which would yield a minimum bit-rate for the state-prediction encoder of

$$r_{\text{SPE}} = \frac{1}{\log 2}\Big(\sum_{i:\ell_i\, e^{\lambda_i[A]T_s} > \gamma(\ell_i - n_i) - w_i} \lambda_i[A] + \frac{1}{T_s} \log \frac{\ell_i}{\gamma(\ell_i - n_i) - w_i} \Big). \tag{3.15}$$

Thus for $\gamma = 1$ and $w_i + n_i$ much smaller then ℓ_i, we essentially recover the minimum bit-rate derived in Theorem 1. This indicates that, at least under low noise, this type of encoding scheme is not overly conservative.

The role of the constant γ can be understood from equation (3.14), which shows that smaller values of γ will lead to a smaller error e between the process' state x and its estimate $\hat{x}$. This in turn will lead to smaller values for x because of (3.8). This means that small values of γ lead to better steady-state performance. The price paid for a small γ is a larger minimum bit-rate r_{SPE} required by this encoding scheme (cf. (3.15)).

Proof of Theorem 6. Since $\hat{x}^-(t_0) = 0$, we have that $e^-(t_0) = x(t_0)$ and therefore

$$|e_i^-(t_0) + \mathbf{n}_i(t_0)| = |x_i(t_0) + \mathbf{n}_i(t_0)| \leq |x_i(t_0)| + |\mathbf{n}_i(t_0)| \leq \ell_i.$$

From this, (3.9), and (3.11), we conclude that

$$|e_i(t)| \leq e^{\lambda_i[A]t} \ell_i \Big| \Delta q_{K_i}\Big(\frac{e_i^-(t_0) + \mathbf{n}_i(t_0)}{\ell_i} \Big) \Big| + |\mathbf{w}_i(t, t_0)| \leq \frac{\ell_i\, e^{\lambda_i[A]t}}{K_i} + w_i, \qquad \forall t \in [t_0, t_1),$$

where $t_1 := t_0 + T_s$ denotes the next sampling time. Since $\frac{\ell_i\, e^{\lambda_i[A]T_s}}{K_i} + w_i \leq \gamma(\ell_i - n_i)$, we further conclude that $|e_i(t)| \leq \gamma(\ell_i - n_i)$, $\forall t \in [t_0, t_1)$. Since $\gamma \leq 1$, equation (3.14) follows by induction. Once it has been established through (3.14) that e remains bounded, one concludes that x also remains bounded because of (3.8). ■

Remark 7. The results above still hold with small modifications when A has complex conjugate eigenvalues. In case $\lambda_i[A]$, $\lambda_j[A]$ is a complex conjugate pair, it would be enough, e.g., to quantize the real part of x_i and the complex part of x_j because both x_i and x_j could be reconstructed from these. Since in a sampling interval both the real and imaginary parts grows as fast as $e^{T_s \Re\lambda_i[A]}$, (3.12) should be replaced by

$$K_i = K_j \geq \frac{\ell_i\, e^{T_s \Re\lambda_i[A]}}{\gamma(\ell_i - n_i) - w_i}, \qquad\qquad \gamma(\ell_i - n_i) \geq w_i.$$

3.2 Variable-step quantization

The quantizer proposed in the previous section may lead to a poor steady-state response if one is forced to choose the ℓ_i large to deal with large initial conditions. Indeed, if $|x_i(t_0)|$ is large, because of (3.13) we need to choose ℓ_i large and therefore the upper bound on e_i given by (3.14) is also large. In practice, and because of (3.10), this may lead to large values for x. We propose to use variable-step quantization to overcome this difficulty.

To construct a variable-step quantizer we start with a sequence of scaling vectors $\{\ell(k) \in (0,\infty)^n : k = 1, 2, \dots\}$ and use the quantizer $Q_{K,\ell(k)}$ at the kth sampling time. To improve the steady-state response, the entries of the scaling vectors $\ell(k)$ should increase as e becomes smaller, leading to better resolution and smaller quantization errors. The number of quantization steps and the sampling rate remains constant and therefore the bit-rate also does not change. The following theorem provides a variable-step quantizer that also guarantees boundedness of the closed-loop system. This quantizer has the desirable feature that even in the presence of noise it allows for a bit-rate arbitrarily close to the minimum one that was derived in Theorem 1 for the noiseless case.

Theorem 8 (Bit-allocation for variable-step quantization). *Let $w_i \geq 0$ and $n_i \geq 0$ denote upper bounds on the absolute values of the ith component of $\mathbf{w}(t,\tau)$, $\tau + T_s \geq t \geq \tau \geq 0$ and $\mathbf{n}(t)$, $t \geq 0$, respectively, and $\gamma \in (0,1]$, $K \in \mathbb{N}^n_{>0}$, $\ell(0) \in (0,\infty)^n$ be such that*

$$K_i > \gamma^{-1} e^{\lambda_i[A]T_s}, \qquad \forall i \in \{1,\dots,n\},$$

and consider a variable-step quantizer constructed using the following sequence of scaling vectors

$$\ell_i(k+1) = \frac{e^{\lambda_i[A]T_s}}{\gamma K_i}\ell_i(k) + \frac{1}{\gamma}w_i + n_i, \qquad k = 0, 1, \dots \tag{3.16}$$

Initializing the encoder and decoder with $\hat{x}^-(t_0) = 0$, where t_0 denotes the first sampling time, and assuming that

$$|x_i(t_0)| \leq \ell_i(0) - n_i, \qquad \forall i \in \{1,\dots,n\}, \tag{3.17}$$

we have that

$$|e_i(t)| \leq \gamma(\ell_i(k) - n_i), \qquad \forall k \geq 1,\ t \in [t_k, t_{k+1}),\ i \in \{1,\dots,n\}, \tag{3.18}$$

and the process' state x remains uniformly bounded.

Before proving Theorem 8, one should note that the minimum bit-rate for the variable-step quantizer proposed is

$$r := \frac{\log \prod_{i:K_i>1} K_i}{T_s \log 2} > \frac{1}{\log 2}\Big(\sum_{i:e^{\lambda_i[A]T_s} \geq \gamma} \lambda_i[A] - \frac{\log \gamma}{T_s}\Big). \tag{3.19}$$

Therefore, by choosing $\gamma = 1$, this quantizer is actually able to achieve any bit-rate above the minimum one that was derived in Theorem 1 for the noiseless case. Moreover, for $\gamma = 1$ the sampling interval T_s does not affect the minimum bit-rate required by the quantizer.

With respect to the steady-state performance, we conclude that $\lim_{k\to\infty} \ell_i(k) = \frac{\frac{1}{\gamma}w_i+n_i}{1-\frac{e^{\lambda_i[A]T_s}}{\gamma K_i}}$ and therefore $|e_i(t)|$ is asymptotically bounded by

$$\gamma\Big(\frac{\frac{1}{\gamma}w_i+n_i}{1-\frac{e^{\lambda_i[A]T_s}}{\gamma K_i}}-n_i\Big)=\frac{w_i+n_i\frac{e^{\lambda_i[A]T_s}}{K_i}}{1-\frac{e^{\lambda_i[A]T_s}}{\gamma K_i}}.$$

Not surprisingly, to achieve the minimum bit-rate, we would need to choose $\gamma = 1$ and the K_i arbitrarily close to $e^{\lambda_i[A]T_s}$, which would lead to a very large upper bound on $\|e\|$ and consequently on $\|x\|$. On the opposite extreme, we can choose K_i much larger than $\frac{e^{\lambda_i[A]T_s}}{\gamma}$, which results in a bit-rate much larger than the minimal one but a smaller upper bound for $\|e\|$ and $\|x\|$. The best achievable upper bound for each $|e_i|$ is w_i.

Remark 9. In the absence of noise and disturbances $n_i = w_i = 0$, all the $\ell_i(k)$ converge to zero exponentially fast and therefore so does e and the process' state x. This means that, for the noiseless case, we can actually have exponential convergence to zero for any bit-rate arbitrarily close to the minimum one derived in Theorem 1.

Proof of Theorem 8. We shall prove by induction that

$$|e_i^-(t_k)| \le \ell_i(k) - n_i, \qquad \forall i \in \{0,\dots,n\}. \tag{3.20}$$

Equation (3.17) provides the basis for induction because $e_i^-(t_0) = x_i(t_0)$. Assuming then that (3.20) holds for some $k \ge 0$, we conclude that $|e_i^-(t_k) + \mathbf{n}_i(t_k)| \le |e_i^-(t_0)| + |\mathbf{n}_i(t_0)| \le \ell_i(k)$. From this, (3.9), and (3.11), we obtain

$$|e_i(t)| \le e^{\lambda_i[A]t}\ell_i(k)\Big|\Delta q_{K_i}\Big(\frac{e_i^-(t_k)+\mathbf{n}_i(t_k)}{\ell_i(k)}\Big)\Big| + |\mathbf{w}_i(t,t_0)| \le \frac{\ell_i(k)\,e^{\lambda_i[A]t}}{K_i} + w_i, \quad \forall t \in [t_k, t_{k+1}),$$

where $t_{k+1} := t_k + T_s$ is the next sampling time. From this and (3.16) we further conclude that

$$|e_i(t)| \le \gamma(\ell_i(k+1) - n_i), \qquad \forall t \in [t_k, t_{k+1}). \tag{3.21}$$

This finishes the induction argument since $\gamma \le 1$ and therefore (3.20) holds for $k+1$. Moreover, because of (3.21) we actually conclude that (3.18) holds. This essentially finished the proof since boundedness of e (and therefore of x) now follows from the fact that the $\ell_i(k)$ are uniformly bounded. ■

4 Examples and experimental results

As discussed in the introduction, the preceding sections served to establish several important facts that have direct applicability to the design of practical coding systems. In this section we present examples to illustrate the usefulness of these results.

Let us consider first a process with a one-dimensional state-space model given by

$$x[k+1] = 1.6x[k] + u[k] + \mathbf{d}[k], \qquad y[k] = x[k] + \mathbf{n}[k].$$

Under no bandwidth constrains, we would apply a control input $u[k] = -Ky[k]$, where K is the optimal control gain that would minimize a given quadratic cost function. Our results compare

Compression Scheme	Quantization Levels	$10^3 \sum x^2$	$10^3 \sum u^2$
None	∞	2.379	1.8041
USQ	16	13.605	1.8756
Standard PC	16	2.523	1.9374
Optimal PC	16	2.410	1.8324
USQ	18	3.497	1.8527
VS-USQ	18	2.462	1.8657
PC-VS-USQ	18	2.305	1.8954

Table 1: Comparison of different encoding schemes.

the performance of six different forms of transmission of the observations $y[k]$: (i) no compression, (ii) uniform scalar quantization (USQ) with constant step size, (iii) predictive coding with simple prediction based on previous observations (Standard PC),(iv) predictive coding with optimal state prediction (Optimal PC), (v) variable step size scalar quantization without prediction (VS-USQ), and (vi) variable step size scalar quantization with prediction (PC-VS-USQ).

In all cases, the overall cost function was chosen to be $\sum_0^\infty (x^2 + u^2)$ over the trajectory, starting from an arbitrary initial condition of $x_0 = 20$. Pseudo-random noise processes $\mathbf{d}$ and $\mathbf{n}$ were simulated as white Gaussian with variance 1. The constant γ in Theorems 6 and 8 was chosen to be 0.1. Following Theorem 6, the minimum bit-rate for this system is 4 bits/sample. Indeed it was observed that the system does not converge for a quantization rate of 3 bits/sample. Results for convergence cost (measured by $\sum x^2$) and control cost (measured by $\sum u^2$) were averaged over 1000 simulations for each case, and are shown in Table 4.

Several observations can be made on the results. First, we can see how the selection of the predictor based on knowledge of the system, as described in Section 3 results in a lower cost. In optimal PC, the aim is to make the best possible prediction of the next input sample, given the previous sample and the system dynamics. In our system, the best predictor of $x[k]$ given $x[k-1]$ would be $\hat{x}[k] = (A + BK)x[k-1]$, where $A = 1.6, B = 1$ and K is the optimal control gain. The difference between this prediction and the actual input sample, $x[k] - \hat{x}[k]$, is then quantized with a fixed uniform quantizer. At the decoder, we receive $Q(e[k])$, and we update the current estimate of the state as $\hat{x}[k] = Q(e[k]) + \hat{x}[k-1]$. Both encoder and decoder retain a one-step memory of the state. To synchronize the encoder and decoder, we work with the state estimate rather than the actual state at every step. In the standard predictive coder, we simply have $\hat{x}[k] = x[k-1]$. We find that the optimal PC scheme has a lower convergence and control cost than the standard PC, since it makes use of the system dynamics to make a better prediction. It also outperforms the non-predictive USQ scheme; since the prediction error has a lower dynamic range than the actual observations, we get better performance with the predictive approach.

A second observation is that VS-USQ can outperform standard USQ. To illustrate this, we simulate a variable step-size quantizer following the adaptation schedule in (3.16). Given a fixed number of levels K_1 (corresponding to a bit-rate greater than the minimum one), the quantizer starts with a large dynamic range $\ell(0)$ for the input, and progressively scales it down, as $\ell(k+1) = \frac{A}{\gamma K_1}\ell(k)$. In our simulations, we assumed $K_1 = 18$, $\ell(0) = 50$, and stopped scaling the step size at

Compression Scheme	Quantization Levels	$10^3 \sum(x_1^2)$	$10^3 \sum(x_2^2)$	$10^3 \sum(u^2)$
None	∞	2.0447	3.4577	1.837
USQ	8, 8	1338.0	3.6	1.9245
USQ	16, 4	212.53	3.68	1.9249
USQ	18, 6	120.02	3.59	1.882
Optimal PC	16, 4	29.899	3.549	1.9223
VS-USQ	18, 6	2.3138	4.0995	2.295
PC-VS-USQ	18, 6	2.1587	3.8339	2.3795

Table 2: Comparison of quantization schemes for the two-dimensional system.

$\ell_{final} = 6$. Clearly, we do better by dynamically adapting the quantizer to the dynamic range of the input. Note that an adaptive quantization will be needed when the initial conditions of the system are allowed to be arbitrary [1]. Also note that unlike well known adaptive quantization techniques used for source coding, here the adaptation is driven by the knowledge of the system structure. Thus, a system with known equations to which a given amount of control has been applied, can be expected to be with high likelihood within a certain distance of the origin, and therefore a choice of step size that is well suited to the current state of the system is possible.

In a second experiment we consider the two-dimensional process given by

$$x[k+1] = \left[\begin{smallmatrix} 1.6 & 0 \\ 0 & .4 \end{smallmatrix}\right] x[k] + \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right] (u[k] + \mathbf{d}[k]), \qquad y[k] = x[k] + \mathbf{n}[k].$$

As before, we consider an ideal control input $u[k] = -Ky[k]$. We use this experiment to illustrate how in the case of multidimensional state variables, one should use more bits to encode those dimensions that are more important for convergence. The eigenvalues of A are simply 1.6 and 0.4. Given $\gamma = 0.1$, the optimal bit-allocation scheme should quantize the first element of the observation vector with at least 4 bits, and the second with at least 2 bits. Any other bit-allocation would perform worse – and this is verified by our results for USQ with 3 bits each, and USQ with 4 and 2 bits respectively. If we allow a bit-rate greater than the minimum, we find a variable step-size scheme (with $\ell(0)$ and ℓ_{final} as in the one-dimensional case) outperforms the fixed step size scheme.

5 Conclusions

We addressed the problem of determining the minimum bit-rate needed to stabilize a linear time-invariant process. In particular, we determined a bit-rate below which stabilization is not possible and above which asymptotic stabilization can be achieved.

Inspired by DPCM, we proposed encoding/decoding schemes that guarantees boundedness of the state for the case of a noisy linear time-invariant process. With fixed-step quantization, we are only able to approach the minimum bit-rate in the noiseless case. However, with variable-step quantization we are able to approach it even in the presence of noise and disturbances.

We assumed that the state of the process was accessible (or could be estimated by the encoder) and was transmitted digitally. We are currently exploring transmitting the process output or the

control signals instead. Another topic of future research is the development of Adaptive DPCM encoding schemes to be used when bounds on the noise are not known a priori.

We considered a purely deterministic framework, where boundedness was required for *every* initial condition and noise/disturbance (inside pre-specified bounded sets). It is worth it to explore how the results derived here would change if one were working in a stochastic setting by attaching probability distributions to the initial conditions/noise/disturbance and relax stability to an average sense. One would then expect that entropy-like coding (characterized by a variable number of bits per symbol) would lead to lower bit-rates.

Acknowledgments

We would like to thank Prof. Sanjoy Mitter for inspiring this line of research through his plenary talk at the 2001 European Control Conf. [5] and later for bringing to our attention the PhD thesis of Tatikonda [8].

References

[1] R. W. Brockett and D. Liberzon. Quantized feedback stabilization of linear systems. *IEEE Trans. on Automat. Contr.*, 45(7), July 2000.

[2] N. Elia and S. K. Mitter. Stabilization of linear systems with limited information. *IEEE Trans. on Automat. Contr.*, 46(9):1384–1400, Sept. 2001.

[3] N. S. Jayant and P. Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[4] D. Liberzon. On stabilization of linear systems using coding and limited communication. Submitted to publication., 2002.

[5] S. K. Mitter. Control with limited information. *European J. Contr.* Special issue with the plenary addresses presented at the European Contr. Confer., 7(2–3):122–131, 2001.

[6] G. N. Nair and R. J. Evans. Communication-limited stabilization of linear systems. In *Proc. of the 39th Conf. on Decision and Contr.*, volume 1, pages 1005–1010, Dec. 2000.

[7] I. R. Petersen and A. V. Savkin. Multi-rate stabilization of multivariable discrete-time linear systems via a limited capacity communication channel. In *Proc. of the 40th Conf. on Decision and Contr.*, volume 1, pages 304–309, Dec. 2001.

[8] S. Tatikonda. *Control Under Communication Constrains.* PhD thesis, MIT, Cambridge, MA, 2000.

[9] W. S. Wong and R. W. Brockett. Systems with finite communication bandwidth–part I: State estimation problems. *IEEE Trans. on Automat. Contr.*, 42(9), Sept. 1997.

[10] W. S. Wong and R. W. Brockett. Systems with finite communication bandwidth–II: Stabilization with limited information feedback. *IEEE Trans. on Automat. Contr.*, 44(5), May 1999.