Control with minimal cost-per-symbol encoding and quasi-optimality of event-based encoders Technical Report

Justin Pearson, João P. Hespanha*, Daniel Liberzon†

May 11, 2016

Abstract

We consider the problem of stabilizing a continuous-time linear time-invariant system subject to communication constraints. A noiseless finite-capacity communication channel connects the process sensors to the controller/actuator. The sensor's state measurements are encoded into symbols from a finite alphabet, transmitted through the channel, and decoded at the controller/actuator. We suppose that the transmission of each symbol costs one unit of communication resources, except for one special symbol in the alphabet that is "free" and effectively signals the absence of transmission. We explore the relationship between the encoder's average bit-rate, its average consumption of communication resources, and the ability of the controller and encoder/decoder pair to stabilize the process. We present a necessary and sufficient condition for the existence of a stabilizing controller and encoder/decoder pair, which depends on the encoder's average bit-rate, its average resource consumption, and the unstable eigenvalues of the process. Moreover, if this condition is satisfied, a stabilizing encoding scheme can be constructed that consumes resources at an arbitrarily small rate, provided the encoder has access to a sufficiently precise clock or large memory. The paper concludes with the analysis of a simple emulation-based controller and event-based encoder/decoder pair that are easy to implement, stabilize the process, and have average bit-rate and resource consumption within a constant factor of the optimal bound.

*J. Pearson and J. P. Hespanha are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, U.S.A., {jppearson,hespanha}@ece.ucsb.edu. This material is based upon work supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-D-0001 from the U.S. Army Research Office.

†D. Liberzon is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, U.S.A., liberzon@uiuc.edu. Supported by NSF grants CNS-1217811 and ECCS-1231196.

1 Introduction

We consider the problem of stabilizing a continuous-time linear time-invariant process subject to communication constraints. The basic setup, also considered in [3, 6, 11, 14, 15, 21] and many other works, assumes that a finite capacity communication channel connects the process sensors to the controller/actuator. An encoder at the sensor sends a symbol through the channel once per sampling time, and the controller determines the actuation signal based on the incoming stream of symbols. The question arises: what is the smallest channel average bit-rate for which a given process can be stabilized? It was shown in [6, 14, 21] that a necessary and sufficient condition for stability can be expressed as a simple relationship between the unstable eigenvalues of the open-loop system matrix and the bit-rate of the communication channel. Extensions of this result have been enthusiastically explored, see [12, 13] and references therein.

A starting point for the present work is the observation that an encoder can effectively save communication resources by occasionally not transmitting information — the absence of an explicitly transmitted symbol nevertheless conveys information. We formulate a framework to capture this by supposing that each symbol's transmission costs one unit of communication resources, except for one special free symbol that represents the absence of a transmission.

Within this framework, we define an encoder's *average cost per symbol* – essentially the largest average fraction of non-free symbols emitted by that encoder over all possible symbol streams. This paper's first technical contribution is a necessary and sufficient condition for the existence of a stabilizing controller and encoder/decoder pair obeying a constraint on its average cost per symbol. This condition depends on the channel's average bit-rate, the encoder's average cost per symbol, and the unstable eigenvalues of the open-loop system matrix. The proof is constructive in that it explicitly provides a family of controllers and encoder/decoder pairs that stabilize the process when the condition holds. The pairs are optimal in the sense that they satisfy the stability condition as tightly as desired. As the constraint on the average cost per symbol is allowed to increase (becomes looser), our necessary and sufficient condition recovers the condition from [6]. Moreover, we show that if an encoder can stabilize the process, then it can do so using arbitrarily small amounts of communication resources per time unit. One way to achieve this is by transmitting only a few non-free symbols per time unit, but being very selective about which transmission period to send them in. Alternatively, the encoder and decoder could share a massive symbol library so that each symbol carries sufficient information about the state. Finally, a counterintuitive corollary to our main result shows that if the process may be stabilized with average bit-rate r bits per time unit, then there exists a stabilizing controller and encoder/decoder pair using average bit-rate r which uses no more than 50% non-free symbols in any stream of symbols it may transmit.

It is important to point out that in our problem setup, the transmission times are fixed; this prevents the encoder from communicating an infinite amount of information in the (real-valued) transmission times, which would require clocks with infinite

precision.

The encoders developed in the first part of the paper are optimal in the sense that they can stabilize a process with an average cost-per-symbol as low as possible. However, they are possibly very complex and difficult to implement. In particular, as an encoder's cost-per-symbol approaches the minimum bound, its codeword library grows to infinite size. In the last part of the paper, we develop an easily-implementable event-based encoder/decoder and compare it to the encoders from the paper's first part.

Recent results in event-based control [1, 2, 10, 18] indicate that an encoder can conserve communication resources by transmitting only on a "need-to-know" basis. Since our framework forces transmission at fixed transmission times, it would appear to prohibit any sort of event-based control. However, our framework can be regarded as event-based if one interprets non-free symbols as transmission-worthy events and the free symbol as "no transmission."

Preliminary work in event-based control assumed that the event-detector could transmit infinite-precision quantities across the communication channel to the controller/actuator. To extend this work to finite-bit-rate communication channels, recent works explore event-based *quantized* control, typically introducing an encoder/decoder or quantizer in the communication path to limit the number of bits transmitted. Several recent works offer strategies for event-based quantized control that study trade-offs between quantizer complexity, bit-rate, and minimum inter-transmission intervals. For example, [7] explores an intuitive event-based quantized control scheme that sends single bits based on the state estimation error transitioning between quantization levels. The design in [8] of an event-based quantized control scheme for a disturbed, stable LTI system allows the state trajectory to match as closely as desired the state-feedback state trajectory that would be obtained without communication constraints. In [19] the authors consider the simultaneous co-design of the event-generator and quantizer for the control of a non-linear system using the hybrid system framework from [5]. Sufficient bit-rates for event-triggered stabilizability of nonlinear systems were also studied in [9]. In [20] a method is developed for event-based quantized control design that achieves a desired convergence rate of a Lyapunov function of the state, while guaranteeing a positive lower bound on inter-transmission times and a uniform upper bound on the number of bits in each transmission.

In contrast to the optimal encoders introduced in the paper's first part, the proposed event-based encoders are easy to implement but not optimal. However, they are only slightly sub-optimal. Specifically, the paper's second technical contribution presents a sufficient condition for the existence of an emulation-based controller and event-based encoder/decoder pair. The condition resembles the sufficient condition from the paper's first part, and exceeds it by less than a factor of 2.5, meaning that the proposed event-based encoding scheme needs at most 2.5 times as many communication resources as an optimal encoding scheme requires. This establishes that event-based encoding schemes can offer "order-optimal" performance in communication-constrained control problems.

The remainder of this paper is organized as follows. Section 3 contains a necessary condition for stability, namely that stability is not possible when our condition does not

hold. To prove this result we actually show that it is not possible to stabilize the process with a large class of encoders — which we call M-of-N encoders — that includes all the encoders with average cost per symbol not exceeding a given threshold. Section 4 contains a sufficient condition for stability, showing that when our condition *does* hold, there is an encoder/decoder pair that can stabilize the process. We explicitly construct a possible encoding scheme. Finally, in Section 5 we develop an event-based encoding scheme that stabilizes the process, provided a sufficient condition holds.

A subset of the results in Sections 3 and 4 appeared with an incomplete proof in the conference paper [16]. This paper provides a complete proof and generalizes the problem statement to permit a larger class of encoders with arbitrary transmission times. Preliminary work for the results in Section 5 appeared in the conference paper [17].

2 Problem Statement

Consider a stabilizable linear time-invariant process

$$\dot{x} = Ax + Bu, \qquad x \in \mathbb{R}^n, u \in \mathbb{R}^m, \tag{1}$$

for which it is known that $x(0)$ belongs to a known bounded set $\mathcal{X}_0 \subset \mathbb{R}^n$. A sensor that measures the state $x(t)$ is connected to the actuator through a finite-data-rate, error-free, and delay-free communication channel, see Figure 1. An *encoder* collocated with the sensor samples the state at a fixed sequence of *transmission times* $\{t_k \in [0,\infty) : k \in \mathbb{N}_{>0}\}$, and from the corresponding sequence of measurements $\{x(t_k) : k \in \mathbb{N}_{>0}\}$ causally constructs a sequence of symbols $\{s_k \in \mathcal{A} : k \in \mathbb{N}_{>0}\}$ from a nonempty finite alphabet $\mathcal{A}$. Without loss of generality, $\mathcal{A} = \{0, 1, \ldots, S\}$ with $S \coloneqq |\mathcal{A}| - 1$. At time t_k the encoder sends the symbol s_k through the channel to a *decoder/controller* collocated with the actuator, which causally constructs the control signal $u(t)$, $t \geqslant 0$ from the sequence of symbols $\{s_k \in \mathcal{A} : k \in \mathbb{N}_{>0}\}$ that arrive at the decoder. The sequence of transmission times $\{t_k\}$ is assumed to be monotonically nondecreasing and unbounded (i.e., $\lim_{k\to\infty} t_k = +\infty$). The fact that the sequence of transmission times is fixed *a priori* prevents the controller from communicating information in the transmission times themselves. Note that because the sequence of transmission times is not necessarily strictly increasing, this allows multiple transmissions at a single time instant, which can be viewed as encoding several symbols in the same message. The non-negative *average bit-rate* r of a sequence of symbols $\{s_k\} \subset \{0, \ldots, S\}$ transmitted at times $\{t_k\}$ is the rate of transmitted information in units of bits per time unit, and is defined as

$$r \coloneqq \log_2(S+1) \limsup_{k\to\infty} \frac{k}{t_k}. \tag{2}$$

We assume that the symbol $0 \in \mathcal{A}$ can be transmitted without consuming any communication resources, but the other S symbols each require one unit of communication resources per transmission. One can think of the "free" symbol 0 as the absence of an

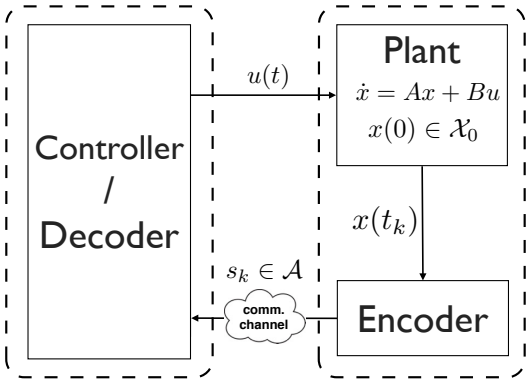

Figure 1: The limited-communication setup. At time t_k, the encoder samples the plant state $x(t_k)$ and selects symbol s_k from alphabet $\mathcal{A}$ to send to the decoder/controller. The decoder/controller constructs the actuation signal $u(t)$ for the plant.

explicit transmission. The "communication resources" at stake may be energy, time, or any other resource that may be consumed in the course of the communication process. In order to capture the average rate at which an encoder consumes communication resources, we define the *average cost per symbol* of an encoder as follows: We say an encoder has *average cost per symbol not exceeding* γ if there exists a non-negative integer N_0 such that for every symbol sequence $\{s_k\}$ generated by the encoder, we have

$$\frac{1}{N_2} \sum_{k=N_1}^{N_1+N_2-1} I_{s_k \neq 0} \leqslant \gamma + \frac{N_0}{N_2} \quad \forall N_1, N_2 \in \mathbb{N}_{>0}, \tag{3}$$

where $I_{s_k \neq 0} \coloneqq 1$ if the kth symbol is not the free symbol, and 0 if it is. The summation in (3) captures the total resources spent transmitting symbols $s_{N_1}, s_{N_1+1}, \ldots, s_{N_1+N_2-1}$, independent of the symbols' transmission times. Motivating this definition of average cost per symbol is the observation that the lefthand side has the intuitive interpretation of the average cost per transmitted symbol between symbols s_{N_1} and $s_{N_1+N_2-1}$. As $N_2 \to \infty$, which corresponds to averaging over a growing window of symbols, the rightmost term vanishes, leaving γ as an upper bound on the average long-term cost per symbol of the symbol sequence. To illustrate the necessity of the N_0 term, note that without it, any symbol sequence with a nonzero symbol at some index k will violate (3) for any $\gamma \in [0, 1)$ by picking $N_1 \coloneqq k$ and $N_2 \coloneqq 1$; the presence of the N_0 term allows an encoder to have a very small average cost per symbol while still enabling it to transmit long runs of non-free symbols. Note that because the left-hand side of (3) never exceeds 1, every encoder has an average cost per symbol not exceeding c for any $c \geqslant 1$. Also, note that any encoder with average cost per symbol not exceeding $\gamma = 0$ can transmit at most N_0 non-free symbols for all time, making it unsuitable for stabilization. For these two reasons, any encoder of interest will have an average cost per symbol not exceeding some $\gamma \in (0, 1]$.

Whereas the average bit-rate r only depends on the symbol alphabet $\mathcal{A}$ and transmission times $\{t_k\}$, the average cost per symbol of an encoder/decoder pair depends on every possible symbol sequence it may generate, and therefore may in general depend on the encoder/decoder pair, the controller, process (1), and the initial condition $x(0)$.

The specific question considered in this paper is: under what conditions on the average bit-rate and average cost per symbol do there exist a controller and encoder/decoder pair that stabilize the state of process (1)?

3 Necessary condition for boundedness with limited-communication encoders

It is known from [6, 14, 21] that it is possible to construct a controller and encoder/decoder pair that stabilize process (1) with average bit-rate r only if

$$r \ln 2 \geqslant \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A], \tag{4}$$

where ln denotes the base-e logarithm, and the summation is over all eigenvalues of A with nonnegative real part. The following result shows that a larger average bit-rate r may be needed when one poses constraints on the encoder's average cost per symbol γ. Specifically, when $\gamma \geqslant S/(S+1)$ the (necessary) stability condition reduces to (4), but when $\gamma < S/(S+1)$ an average bit-rate r larger than (4) is necessary for stability.

Theorem 1. *Suppose a controller and encoder/decoder pair keep the state of process (1) bounded for every initial condition $x_0 \in \mathcal{X}_0$. If the encoder uses an alphabet $\{0, \ldots, S\}$, has average bit-rate r, and has average cost per symbol not exceeding γ, then we must have*

$$r\, f(\gamma, S) \ln 2 \geqslant \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A], \tag{5}$$

where the function $f : [0,1] \times \mathbb{N}_{>0} \to [0,\infty)$ is defined as

$$f(\gamma, S) \coloneqq \begin{cases} \frac{H(\gamma)+\gamma \log_2 S}{\log_2(S+1)} & 0 \leqslant \gamma < \frac{S}{S+1} \\ 1 & \frac{S}{S+1} \leqslant \gamma \leqslant 1, \end{cases} \tag{6}$$

and $H(p) \coloneqq -p\log_2(p) - (1-p)\log_2(1-p)$ is the base-2 entropy of a Bernoulli random variable with parameter p.

It is worth making three observations regarding the function f: First, $f(\gamma, S)$ is nondecreasing and continuous in γ for any fixed S, as illustrated in Figure 2. Second, $f(\gamma, S)$ is monotone nonincreasing in S for any fixed $\gamma \in [0,1]$. Therefore, for a fixed r and γ, an encoder can increase its value of $f(\gamma, S)$ "for free" by decreasing S while

commensurately decreasing its average transmission period to keep r constant in accordance with (2). This implies that smaller alphabets are preferable to large ones when trying to satisfy (5) with a given fixed average bit-rate and average cost per symbol.

The third observation is that the average *cost per time unit*, which is $\gamma \limsup_{k\to\infty} \frac{k}{t_k}$, can be made arbitrarily small while still satisfying (5). This can be achieved in several ways:

1. *Large symbol library with infrequent transmissions:* For a given average cost per symbol γ, pick the encoder's transmission times as $t_k \coloneqq kT$ for sufficiently large T so that the average cost per time unit $\gamma \limsup_{k\to\infty} k/t_k = \gamma/T$ is as small as desired. Then, using $r \coloneqq \log_2(S+1)/T$ and leveraging the fact that

$$rf(\gamma, S) = \begin{cases} \frac{H(\gamma)+\gamma \log_2 S}{T} & 0 \leqslant \gamma < \frac{S}{S+1} \\ \frac{\log_2(S+1)}{T} & \frac{S}{S+1} \leqslant \gamma \leqslant 1 \end{cases} \tag{7}$$

   is monotone increasing in S for fixed γ, pick S large enough to satisfy (5). By choosing a large T and S, this scheme elects to send data-rich symbols only infrequently. The state — although remaining bounded — may grow quite large between these infrequent transmissions. Moreover, the large symbol library may require sizeable computational resources to store and process.

2. *Large symbol library with costly symbols rarely sent:* If the encoder's transmission times $\{t_k\}$ are fixed, pick γ small enough to make the average cost per time unit $\gamma \limsup_{k\to\infty} k/t_k$ as small as desired, then increase S as in the previous case to satisfy (5). Like the previous case, this approach requires processing a large symbol library.

3. *Frequent transmissions with costly symbols rarely sent:* If the number of non-free symbols S is fixed, it is still possible to choose an average cost per symbol γ and transmission times $t_k \coloneqq kT$ so that (5) is satisfied and the average cost per time unit $\gamma \limsup_{k\to\infty} k/t_k$ is as small as desired. To verify that this is possible, note that the sequences $\gamma_i \coloneqq e^{-i}$, $T_i \coloneqq e^{-i}\sqrt{i}$, $i \in \mathbb{N}_{>0}$

   have the property that as $i \to \infty$, we have $\gamma_i \to 0$, $T_i \to 0$, and $\gamma_i/T_i \to 0$, but $H(\gamma_i)/T_i \to \infty$, so leveraging (7) we conclude that $r_i f(\gamma_i, S) \ln 2 \to \infty$ (where $r_i \coloneqq \log_2(S+1)/T_i$). This means that one can find $i \in \mathbb{N}_{>0}$ sufficiently large to make the average cost per time unit arbitrarily small and also satisfy the necessary condition (5). In practice, to operate with a very small sampling period T, this approach requires an encoder/decoder pair with a very precise clock.

Remark 1. The addition of the "free" symbol effectively increases the average bit-rate without increasing the rate of resource consumption, as seen by the following two observations:

- Without the free symbols, the size of the alphabet would be S and the average bit-rate would be
$$\log_2(S) \limsup_{k\to\infty} \frac{k}{t_k} < \log_2(S+1) \limsup_{k\to\infty} \frac{k}{t_k}.$$
It could happen that this average bit-rate is too small to bound the plant, yet after the introduction of the free symbol, the condition (5) is satisfied.

- Since γ is essentially the fraction of non-free symbols, the quantity $r\gamma$ is the number of bits per time unit spent transmitting non-free symbols. But since $f(\gamma, S) \geqslant \gamma$, again we see that the free symbols help satisfy (5). To see that $f(\gamma, S) \geqslant \gamma$, observe that for any $S \in \mathbb{N}_{>0}$, $f(\cdot, S)$ is concave and reaches 1 before the identity function does, hence it is everywhere above the identity function on $(0, 1)$, and it matches the identity function at the endpoints 0 and 1.

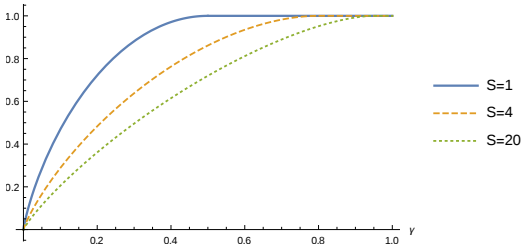

Figure 2: A plot of $f(\gamma, S)$ versus γ for $S = 1, 4, 20$.

3.1 Setup and Proof of Theorem 1

We lead up to the proof of Theorem 1 by first establishing three lemmas centered around a restricted but large class of encoders called M-of-N encoders. We first define M-of-N encoders, which essentially partition their symbol sequences into N-length *codewords*, each with M or fewer non-free symbols. Lemma 1 demonstrates that every encoder with a bounded average cost per symbol is an M-of-N encoder for appropriate N and M. Next, in Lemma 2 we establish a relationship between the number of codewords available to an M-of-N encoder and the function f as defined in (6). Then, in Lemma 3 we establish a necessary condition for an M-of-N encoder to bound the state of process (1). Finally, the proof of Theorem 1 is built upon these three results.

We now introduce the class of M-of-N encoders. For $N \in \mathbb{N}_{>0}$, $\ell \in \mathbb{N}_{\geqslant 0}$, we define the ℓ^{th} *N-symbol codeword* to be the sequence $\{s_{\ell N+1}, s_{\ell N+2}, \ldots, s_{\ell N+N}\}$ of N consecutive symbols starting at the index $k = \ell N + 1$. For $M \in \mathbb{R}_{\geqslant 0}$ with $M \leqslant N$, an *M-of-N encoder* is an encoder for which every N-symbol codeword has M or fewer non-free symbols, i.e.,
$$\sum_{k=\ell N+1}^{\ell N+N} I_{s_k \neq 0} \leqslant M, \qquad \forall \ell \in \mathbb{N}_{\geqslant 0}. \tag{8}$$

The total number of distinct N-symbol codewords available to an M-of-N encoder is thus given by

$$L(N, M, S) \coloneqq \sum_{i=0}^{\lfloor M \rfloor} \binom{N}{i} S^i, \tag{9}$$

where the ith term in the summation counts the number of N-symbol codewords with exactly i non-free symbols. In keeping with the problem setup, the M-of-N encoders considered here each draw their symbols from the symbol library $\mathcal{A} \coloneqq \{0, 1, \ldots, S\}$ and transmit symbols at times $\{t_k\}$.

An intuitive property of M-of-N encoders is that they have an average cost per symbol not exceeding M/N with $N_0 = 2M$. This result is presented as Lemma 5 in the appendix.

The fact that an M-of-N encoder refrains from sending "expensive" codewords effectively reduces its ability to transmit information: A codeword sent from an M-of-N encoder conveys $\log_2 L(N, M, S)$ bits of information, whereas a codeword from an encoder without the M-of-N constraint conveys $\log_2 L(N, N, S) = N \log_2(S + 1)$ bits.

The next lemma, proved in the appendix, shows that the set of M-of-N encoders is "complete" in the sense that every encoder with average cost per symbol not exceeding a finite threshold γ is actually an M-of-N encoder for N sufficiently large and $M \approx \gamma N$.

Lemma 1. *For any encoder/decoder pair with average cost per symbol not exceeding $\gamma \in (0, 1]$, and every constant $\epsilon > 0$, there exist $M \in \mathbb{R}_{\geqslant 0}$ and $N \in \mathbb{N}_{>0}$ with $M < N\gamma(1 + \epsilon)$ such that the encoder/decoder pair is an M-of-N encoder.*

The next lemma establishes a relationship between the number of codewords $L(N, M, S)$ available to an M-of-N encoder and the function f defined in (6).

Lemma 2. *For any $N \in \mathbb{N}_{>0}$, $S \in \mathbb{N}_{\geqslant 0}$ and $\gamma \in [0, 1]$, the function L defined in (9) and the function f defined in (6) satisfy*

$$\frac{\ln L(N, N\gamma, S)}{N \ln(S + 1)} \leqslant f(\gamma, S), \tag{10}$$

with equality holding only when $\gamma = 0$ or $\gamma = 1$. Moreover, we have asymptotic equality in the sense that

$$\lim_{N \to \infty} \frac{\ln L(N, N\gamma, S)}{N \ln(S + 1)} = f(\gamma, S). \tag{11}$$

The left and right sides of (10) are plotted in Figure 3.

Proof of Lemma 2. In this proof we use the base-2 logarithm to match the notation of an information theoretic theorem that we invoke. Let $N \in \mathbb{N}_{>0}$ and $S \in \mathbb{N}_{\geqslant 0}$ be

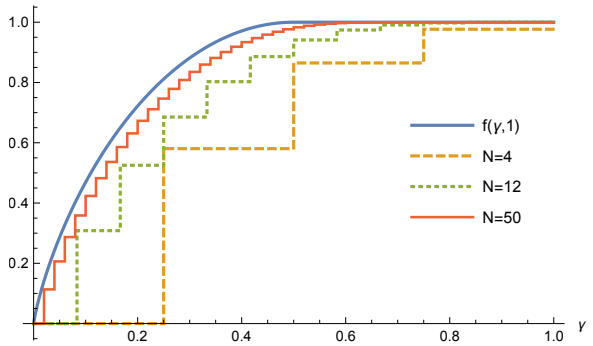

Figure 3: A plot of $f(\gamma, S)$ and $\ln L(N, N\gamma, S)/N \ln(S+1)$ versus γ for $S = 1$ and $N = 4, 12, 50$.

arbitrary. First we prove (10) for $\gamma \in \left(0, \frac{S}{S+1}\right]$. Applying the Binomial Theorem to the identity $1 = (\gamma + (1-\gamma))^N$, we obtain

$$1 = \sum_{i=0}^{N} \binom{N}{i} \gamma^i (1-\gamma)^{N-i}.$$

Since each term in the summation is positive, keeping only the first $\lfloor N\gamma \rfloor$ terms yields the inequality

$$1 > \sum_{i=0}^{\lfloor N\gamma \rfloor} \binom{N}{i} \gamma^i (1-\gamma)^{N-i}. \tag{12}$$

Next, a calculation presented as Lemma 6 in the appendix reveals that

$$\gamma^i (1-\gamma)^{N-i} \geqslant 2^{-N\,H(\gamma)}\, \frac{S^i}{S^{N\gamma}} \tag{13}$$

for all $N, S \in \mathbb{N}_{>0}$, $\gamma \in \left(0, \frac{S}{S+1}\right]$, and $i \in [0, N\gamma]$. Using this in (12) and taking $\log_2$ of both sides yields

$$\frac{\log_2 L(N, N\gamma, S)}{N} < H(\gamma) + \gamma \log_2 S. \tag{14}$$

By the definition of f, we have $\log_2(S+1) f(\gamma, S) = H(\gamma) + \gamma \log_2 S$ when $\gamma \in \left[0, \frac{S}{S+1}\right]$. Thus, (14) proves the strict inequality in (10) for $\gamma \in \left(0, \frac{S}{S+1}\right]$. Next, suppose $\gamma \in \left(\frac{S}{S+1}, 1\right)$ and observe from (9) that $L(N, M, S)$ is a sum of positive terms whose index reaches $\lfloor M \rfloor$, hence $L(N, N\gamma, S)$ is strictly less than $L(N, N, S)$ for any $\gamma < 1$. We conclude that

$$\begin{aligned} \frac{\log_2 L(N, N\gamma, S)}{N} &< \frac{\log_2 L(N, N, S)}{N} \\ &= \log_2(S+1) = \log_2(S+1) f(\gamma, S), \end{aligned} \tag{15}$$

where the first equality follows simply from the fact that $L(N, N, S)$ is the number of all possible codewords of length N and hence equals $(S+1)^N$, and the second equality follows from the definition of f when $\gamma \in (\frac{S}{S+1}, 1)$. This concludes the proof of the strict inequality in (10) for $\gamma \in (0, 1)$. The proof of (10) for $\gamma = 0$ follows merely from inspection of (10), and the $\gamma = 1$ case follows from (15).

Next we prove the asymptotic result (11) using information-theoretic methods. First we prove (11) for $\gamma \in [0, \frac{S}{S+1})$. Consider a random variable X parameterized by $S \in \mathbb{N}_{\geqslant 0}$ and $\gamma \in [0, \frac{S}{S+1})$ which takes values in $\mathcal{X} \coloneqq \{0, 1, \ldots, S\}$ with probabilities given by $\mathbb{P}(X = 0) \coloneqq (1 - \gamma)$ and $\mathbb{P}(X = i) \coloneqq \gamma/S$, $i = \{1, 2, \ldots, S\}$. Following our convention, we call 0 the "free" symbol and $1, \ldots, S$ the "non-free" symbols. To lighten notation we write $p(x) \coloneqq \mathbb{P}(X = x), x \in \mathcal{X}$. The entropy of the random variable X is

$$H(X) \coloneqq -\sum_{i=0}^{S} p(i) \log_2 p(i) = H(\gamma) + \gamma \log_2 S, \tag{16}$$

where we have overloaded the symbol H so that $H(\gamma) \coloneqq -\gamma \log_2 \gamma - (1-\gamma) \log_2 (1-\gamma)$ is the entropy of a Bernoulli random variable with parameter γ.

Next, for some arbitrary $N \in \mathbb{N}_{>0}$, we consider N-length sequences of i.i.d. copies of X. Let $\mathcal{X}^N \coloneqq \{(x_1, \ldots, x_N) : x_i \in \mathcal{X}\}$. We use the symbol x^N as shorthand for $(x_1, \ldots, x_N)$, and we use $p(x^N)$ as shorthand for $\mathbb{P}\Big((X_1, X_2, \ldots, X_N) = (x_1, x_2, \ldots, x_N)\Big)$.

Given an N-length sequence $x^N \in \mathcal{X}^N$, the probability that the N i.i.d. random variables $(X_1, \ldots, X_N)$ take on the values in the N-tuple x^N is given by

$$p(x^N) = (1-\gamma)^{N - \sum_{i=1}^N I_{x_i \neq 0}} \; \frac{\gamma^{\sum_{i=1}^N I_{x_i \neq 0}}}{S^{\sum_{i=1}^N I_{x_i \neq 0}}}. \tag{17}$$

The summation $\sum_{i=1}^N I_{x_i \neq 0}$ is the number of non-free symbols in the N-tuple x^N.

For arbitrary $\epsilon > 0$, define the set $A_\epsilon^{(N)} \subseteq \mathcal{X}^N$ as

$$A_\epsilon^{(N)} \coloneqq \left\{ x^N \in \mathcal{X}^N \;\middle|\; N(\gamma - \delta_\epsilon) \leqslant \sum_{i=1}^N I_{x_i \neq 0} \leqslant N(\gamma + \delta_\epsilon) \right\}, \tag{18}$$

where $\delta_\epsilon \coloneqq \epsilon / \log_2 \frac{(1-\gamma)S}{\gamma}$. That is, $A_\epsilon^{(N)}$ is the set of all N-length sequences with "roughly" $N\gamma$ non-free symbols. Using (16), (17), and the definition of δ_ϵ, we can express the inequalities in (18) as

$$A_\epsilon^{(N)} = \left\{ x^N \in \mathcal{X}^N \;\middle|\; 2^{-N(H(X)+\epsilon)} \leqslant p(x^N) \leqslant 2^{-N(H(X)-\epsilon)} \right\}. \tag{19}$$

Here we relied on the fact that $\frac{(1-\gamma)S}{\gamma} > 1$ for $S \in \mathbb{N}_{>0}, \gamma \in [0, \frac{S}{S+1})$. In the form of (19), we recognize $A_\epsilon^{(N)}$ as the so-called "typical set" of N-length sequences of i.i.d. copies of

X as defined in [4]. Theorem 3.1.2 of [4] uses the Asymptotic Equipartition Property of sequences of i.i.d. random variables to prove that for any $\epsilon > 0$, we have

$$(1-\epsilon)2^{N(H(X)-\epsilon)} \leqslant \left|A_\epsilon^{(N)}\right| \tag{20}$$

for $N \in \mathbb{N}_{>0}$ large enough.
Next, we observe that

$$\left|A_\epsilon^{(N)}\right| \leqslant L\left(N, N\left(\gamma+\delta_\epsilon\right), S\right), \tag{21}$$

because $|A_\epsilon^{(N)}|$ is the number of N-length sequences with a number of non-frees in the interval $[N(\gamma-\delta_\epsilon), N(\gamma+\delta_\epsilon)]$, whereas the right-hand side counts sequences with a number of non-frees in the larger interval $[0, N(\gamma+\delta_\epsilon)]$. Combining (20) and (21), we obtain that for any $\epsilon > 0$,

$$\begin{aligned}\frac{1}{N}\log_2(1-\epsilon) + H(\gamma) + \gamma\log_2 S - \epsilon \leqslant \\ \frac{1}{N}\log_2 L\left(N, N\left(\gamma+\delta_\epsilon\right), S\right)\end{aligned} \tag{22}$$

for N large enough. Moreover, by (10) we have

$$\begin{aligned}\frac{1}{N}\log_2 L\left(N, N\left(\gamma+\delta_\epsilon\right), S\right) \leqslant \\ H\left(\gamma+\delta_\epsilon\right) + \left(\gamma+\delta_\epsilon\right)\log_2 S\end{aligned} \tag{23}$$

for any $\gamma \in [0, \frac{S}{S+1})$, $N, S \in \mathbb{N}_{>0}$, and $\epsilon > 0$. Combining these two observations establishes an upper and lower bound on $\frac{1}{N}\log_2 L\left(N, N\left(\gamma+\delta_\epsilon\right), S\right)$. Letting $\epsilon \to 0$, the upper and lower bounds converge to $H(\gamma) + \gamma\log_2 S$, establishing (11) for $\gamma \in [0, \frac{S}{S+1})$. Since the upper and lower bounds are continuous in γ, this proves (11) for $\gamma = \frac{S}{S+1}$ as well.
Lastly, suppose $\gamma \in (\frac{S}{S+1}, 1]$. Since L is monotonically nondecreasing in its second argument, we have

$$\frac{1}{N}\log_2 L\left(N, N\frac{S}{S+1}, S\right) \leqslant \frac{1}{N}\log_2 L\left(N, N\gamma, S\right). \tag{24}$$

Moreover, by (10) we have

$$\frac{1}{N}\log_2 L\left(N, N\gamma, S\right) \leqslant \log_2(S+1). \tag{25}$$

Combining these establishes an upper and lower bound on $\frac{1}{N}\log_2 L\left(N, N\gamma, S\right)$. Taking $N \to \infty$, the bounds become equal because (11) holds for $\gamma = \frac{S}{S+1}$ in the lower bound. Here we relied on the fact that $f(\gamma, S)$ is continuous in γ. We obtain

$$\lim_{N\to\infty}\frac{1}{N}\log_2 L\left(N, N\gamma, S\right) = \log_2(S+1). \tag{26}$$

This concludes the proof of Lemma 2. ∎

The following lemma provides a necessary condition for an M-of-N encoder to be able to bound the state of process (1).

Lemma 3. *Consider an M-of-N encoder/decoder pair with average bit-rate r using a channel with alphabet $\{0, \dots, S\}$ (with 0 the free symbol). If the pair keeps the state of process (1) bounded for every initial condition, then we must have*

$$r \, \frac{\ln L(N, M, S)}{N \ln(S+1)} \ln 2 \geqslant \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A]. \tag{27}$$

Proof of Lemma 3. The proof of this result can be constructed using an argument similar to the ones found in [6, 21], which considers the rate at which the uncertainty on the state, as measured by the volume of the set where it is known to lie, grows through the process dynamics (1) and shrinks upon the receipt of each N-symbol codeword.
We proceed with a proof by contradiction inspired by [6, 21], which considers the rate at which the uncertainty on the state, as measured by the volume $\mathcal{X}_0 \subset \mathbb{R}^n$ of the set where the initial state is known to lie, grows through process (1) and shrinks upon the receipt of information from the encoder. Consider an encoder/decoder pair whose encoder is an M-of-N encoder using symbols $\{0, \dots, S\}$ and has average bit-rate r. For the sake of contradiction, suppose the controller and encoder/decoder pair keep the state of process (1) bounded for every initial condition $x_0 \in \mathcal{X}_0$, but that

$$r < r_{\min} \coloneqq \frac{N \ln(S+1)}{\ln L(N, M, S) \ln 2} \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A]. \tag{28}$$

After a change of coordinates, process (1) can be transformed to

$$\begin{bmatrix} \dot{x}_+ \\ \dot{x}_- \end{bmatrix} = \begin{bmatrix} A_+ & 0 \\ 0 & A_- \end{bmatrix} \begin{bmatrix} x_+ \\ x_- \end{bmatrix} + \begin{bmatrix} B_+ \\ B_- \end{bmatrix} u, \tag{29}$$

where $x_+ \in \mathbb{R}^{n_+}$, $x_- \in \mathbb{R}^{n_-}$, $u \in \mathbb{R}^m$, $n_+ + n_- = n$, and the eigenvalues of A are partitioned between $A_+ \in \mathbb{R}^{n_+ \times n_+}$ and $A_- \in \mathbb{R}^{n_- \times n_-}$, with A_+ having the eigenvalues of A with strictly positive real part and A_- the remaining ones. We focus our attention on the unstable subsystem

$$\dot{x}_+ = A_+ x_+ + B_+ u, \qquad x_+ \in \mathbb{R}^{n_+}, u \in \mathbb{R}^m. \tag{30}$$

Let $\varphi_+(t; x_0)$ denote the solution of (30) in closed-loop, that is, where $u(t)$ is determined by the decoder/controller in response to symbols sent by the encoder. Suppose that by time t the decoder/controller has observed the specific sequence of symbols $\{s_k : t_k \leqslant t\}$. Define

$$\begin{aligned} \mathcal{X}_+(t) \coloneqq \big\{ x_+ \in \mathbb{R}^{n_+} : \exists x_0 \in \mathcal{X}_0 : \varphi_+(t; x_0) = x_+ \\ \& \;\; \mathbf{Enc}(t, x_0) = \{ s_k \;:\; t_{\lfloor \frac{k}{N} \rfloor N} \leqslant t \} \big\}, \end{aligned}$$

where $\mathbf{Enc}(t; x_0)$ denotes the set of codewords that the decoder/controller has observed from the encoder over time interval $[0, t]$ as process (1) runs in closed-loop from the initial condition $x(0) = x_0$. The set $\mathcal{X}_+(t)$ is the tightest set of points that the decoder/controller can deduce that the state x_+ lies in at time t, based on the observation of all N-length codewords up to time t. Since the decoder/controller cannot be certain of where $x_+(t)$ lies within $\mathcal{X}_+(t)$, we refer to $\mathcal{X}_+(t)$ as the *uncertainty region*.
Let $\nu(t) := \int_{x \in \mathcal{X}_+(t)} dx$ denote the volume of $\mathcal{X}_+(t)$, and let $\nu^+(t) \coloneqq \lim_{\tau \downarrow t} \nu(\tau)$ and $\nu^-(t) \coloneqq \lim_{\tau \uparrow t} \nu(\tau)$ denote the limits of $\nu(t)$ from above and below.
Let us now explore how the volume of the uncertainty region evolves due to the process. For arbitrary $k \in \mathbb{N}_{>0}$, consider the open time interval $\big(t_{kN}, t_{kN+N}\big)$ during which the kth codeword is transmitted. Since no complete codewords arrive in this time interval, $\mathbf{Enc}(t; x_0)$ remains constant and the set $\mathcal{X}_+(t)$ simply expands under process (30) for $t \in \big(t_{kN}, t_{kN+N}\big)$. By the variation of constants formula,

$$x_+(t_{kN+N}) = e^{A_+(t_{kN+N} - t_{kN})} x_+(t_{kN}) + u_k, \tag{31}$$

where $u_k := \int_{t_{kN}}^{t_{kN+N}} e^{A_+(t_{kN+N} - t_{kN} - \tau)} B_+ u(\tau) d\tau$. Therefore $\mathcal{X}_+(t) = e^{A_+(t - t_{kN})} \mathcal{X}_+(t_{kN}) + u_k$ for $t \in \big(t_{kN}, t_{kN+N}\big)$. The volume $\nu^-(t_{kN+N})$ is then given by

$$\nu^-(t_{kN+N}) = \int_{x \in e^{A_+(t_{kN+N} - t_{kN})} \mathcal{X}_+(t_{kN}) + u_k} dx. \tag{32}$$

Next we define $z := e^{A_+(t_{kN+N} - t_{kN})} x + u_k$ and apply the integral substitution formula

$$\int_{\varphi(U)} g(x)\, dx = \int_U g(\varphi(z))\, |\det(\mathrm{D}\, \varphi)(z)|\, dz \tag{33}$$

with the values $U := \mathcal{X}_+(t_{kN})$, $g(x) := 1$, $\varphi(x) := e^{A_+(t_{kN+N} - t_{kN})} x + u_k$, for which $\mathrm{D}\, \varphi = e^{A_+(t_{kN+N} - t_{kN})}$. This yields

$$\begin{aligned} \nu^-(t_{kN+N}) &= \int_{x \in \mathcal{X}_+(t_{kN})} |\det e^{A_+(t_{kN+N} - t_{kN})}|\, dx & (34) \\ &= |\det e^{A_+(t_{kN+N} - t_{kN})}|\ \nu^+(t_{kN}). & (35) \end{aligned}$$

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 $\lambda_1[M], \lambda_2[M], ..., \lambda_n[M]$, we conclude that

$$\begin{aligned} \nu^-(t_{kN+N}) &= e^{(t_{kN+N} - t_{kN}) \sum_{i=1}^{n_+} \lambda_i[A_+]} \mu(\mathcal{X}_+(t_{kN})) \\ &= e^{(t_{kN+N} - t_{kN}) \sum_{i : \Re \lambda_i[A] > 0} \lambda_i[A]} \nu^+(t_{kN}), \end{aligned} \tag{36}$$

where the second equality follows from the fact that the eigenvalues of A_+ are precisely the eigenvalues of A with positive real part. Equation (36) establishes the rate of expansion of the uncertainty region between codewords.

Next we characterize how much the uncertainty region shrinks upon the receipt of a codeword.

Let $\mathcal{C} \subset \{0, \ldots, S\}^N$ denote the set of N-length codewords with M or fewer non-free symbols, and note that $|\mathcal{C}| = L(N, M, S)$. Consider $\nu^-(t_{kN})$, the volume of the uncertainty region immediately before a codeword is received at time t_{kN}. Depending on precisely which codeword is received, the volume of the uncertainty region may shrink. To capture this, for each codeword $c \in \mathcal{C}$ let $\nu^+(t_{kN}|c)$ denote the volume of the uncertainty region at time t_{kN} supposing that codeword c is received at that time.

Since for every point x' in the pre-codeword uncertainty region there must exist at least one codeword for which x' is in the post-codeword uncertainty region, we must have $\nu^-(t_{kN}) \leqslant \sum_{c\in\mathcal{C}} \nu^+(t_{kN}|c) \leqslant |\mathcal{C}| \max_{c\in\mathcal{C}} \nu^+(t_{kN}|c)$, and so there must exist a codeword $c^* \coloneqq \arg\max_{c\in\mathcal{C}} \nu^+(t_{kN}|c)$ for which $\nu^+(t_{kN}|c^*) \geqslant \frac{1}{|\mathcal{C}|}\, \nu^-(t_{kN}) = \frac{1}{L(N,M,S)}\, \nu^-(t_{kN})$.

Provided that $\nu^-(t_{kN}) > 0$, there exists a set of initial conditions for which the closed-loop solution results in codeword c^* being transmitted at time t_{kN} and therefore

$$\nu^+(t_{kN}) \geqslant \frac{1}{L(N, M, S)} \nu^-(t_{kN}). \tag{37}$$

Thus, there exist initial conditions for which, at time t_{kN}, the post-codeword uncertainty region is at least $1/L(N, M, S)$ times as big as the pre-codeword uncertainty region. In other words, for certain initial conditions, at time t_{kN} the scheme cannot reduce the uncertainty volume by more than a factor of $1/L(N, M, S)$.

Iterating (36) and (37) from time 0 to t_{kN} for arbitrary $k \in \mathbb{N}_{>0}$, we conclude that for appropriately selected initial conditions, we will have

$$\nu^+(t_{kN}) \geqslant \frac{1}{L(N, M, S)^k} e^{t_{kN} \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A]} \nu(0) \tag{38}$$

$$= e^{t_{kN} \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A] - k \ln L(N,M,S)} \nu(0) \tag{39}$$

Next, let us consider the consequences of our limited bit-rate $r < r_{\min}$. Define $\delta \coloneqq r_{\min} - r$ so that $r_{\min} - \delta/2 > r$. Using the definition of r from (2) we find that

$$\frac{r_{\min} - \delta/2}{\log_2(S+1)} > \limsup_{k\to\infty} \frac{k}{t_k}, \tag{40}$$

meaning that $(r_{\min} - \delta/2)/\log_2(S+1)$ is an eventual upper bound on the sequence $\{k/t_k\}$, and therefore also on the sequence $\{kN/t_{kN}\}$. This means that for any $\epsilon > 0$, there exists $K \in \mathbb{N}_{>0}$ such that for all $k > K$ we have

$$\frac{kN}{t_{kN}} < \frac{r_{\min} - \delta/2}{\log_2(S+1)} + \epsilon. \tag{41}$$

In particular, pick $\epsilon = \delta/(4\log_2(S+1))$ in (41). Using the definition of $r_{\min}$ from (28) and straightforward algebraic manipulations yields

$$t_{kN} \left(\frac{\delta \ln L(N, M, S)}{4N \log_2(S+1)} \right) < t_{kN} \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A]$$

$$- k \ln L(N, M, S) \qquad \forall k > K.$$

The left-hand side is unbounded because the sequence $\{t_k\}$ is unbounded. Hence, we conclude that

$$\lim_{k\to\infty} \left(t_{kN} \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A] - k \ln L(N, M, S) \right) = \infty.$$

Note that this is the exponent in (39), which means that the volume of sets $\{\mathcal{X}_+(t)\}$ grows to infinity as $t \to \infty$, which in turn means that we can find values for the state in these sets arbitrarily far apart for sufficiently large t and thus arbitrarily far from the origin. We thus conclude that the controller and encoder/decoder pair cannot stabilize the process. ■

Now we are ready to prove Theorem 1.

Proof of Theorem 1. If $\gamma = 0$, then the encoder transmits at most N_0 non-free symbols, and therefore cannot bound an unstable system for all time. We assumed that the encoding scheme keeps the state of process (1) bounded, so we must have $\sum_{i:\Re\lambda_i[A]>0} \lambda_i[A] = 0$, and so (5) is satisfied trivially. Now suppose $\gamma > 0$. By Lemma 1, for any $\epsilon > 0$ there exist $M \in \mathbb{R}_{\geqslant 0}$ and $N \in \mathbb{N}_{>0}$ with $M < N\gamma(1+\epsilon)$ for which the encoder/decoder is an M-of-N encoder. Since the state of the process is kept bounded, by Lemma 3 we have

$$\sum_{i:\Re\lambda_i[A]>0} \lambda_i[A] \leqslant r \, \frac{\ln L(N, M, S)}{N \ln(S+1)} \, \ln 2. \tag{42}$$

Since L is monotonically nondecreasing in its second argument and $M < N\gamma(1+\epsilon)$, we have

$$r \, \frac{\ln L(N, M, S)}{N \ln(S+1)} \leqslant r \, \frac{\ln L(N, N\gamma(1+\epsilon), S)}{N \ln(S+1)}. \tag{43}$$

Lemma 2 implies that

$$r \, \frac{\ln L(N, N\gamma(1+\epsilon), S)}{N \ln(S+1)} \leqslant r f(\gamma(1+\epsilon), S). \tag{44}$$

Combining these and letting $\epsilon \to 0$, we obtain (5). This completes the proof of Theorem 1. ■

4 Sufficient condition for stability with limited-communication encoders

The previous section established a necessary condition (5) on the average bit-rate and average cost per symbol of an encoder/decoder pair in order to bound process (1). In this section, we show that with a strict inequality this condition is also sufficient for a stabilizing encoder/decoder to exist. The proof is constructive in that we provide the encoder/decoder.

The proposed scheme is sometimes called *emulation-based* because the encoder/decoder emulates a stabilizing state-feedback controller $u = Kx$. This state-feedback controller cannot be used in the limited-communication environment considered in this paper because the infinite-precision state $x(t) \in \mathbb{R}^n$ cannot be sent over the channel and hence is unavailable to the controller. Instead, in emulation-based control, the state-feedback controller is coupled to an encoder/decoder pair that estimates the state as $\hat{x}(t)$, resulting in the control law $u(t) = K\hat{x}(t),\ t \geqslant 0$.

Theorem 2. *Assume that $A+BK$ is Hurwitz. For every $S \in \mathbb{N}_{\geqslant 0}$, $r \geqslant 0$, and $\gamma \in [0,1]$ satisfying*

$$r f(\gamma, S) \ln 2 > \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A], \tag{45}$$

where the function f is defined in (6), there exists an emulation-based controller and an M-of-N encoder/decoder pair that uses S non-free symbols, has average bit-rate not exceeding r, has an average cost per symbol not exceeding γ, and exponentially stabilizes process (1) for every initial condition $x_0 \in \mathcal{X}_0$.

Remark 2. The encoding scheme that follows relies on a strict inequality in (45) for the existence of a suitable M-of-N encoder, and as that gap shrinks to 0, the codeword length N becomes unbounded. In contrast, the event-based encoding scheme presented in Section 5 has the property that if its corresponding data-rate condition (82) holds with equality, the scheme bounds the state of the process, *cf.* Remark 5.

The proof of Theorem 2 uses the following lemma, proved in the appendix, which establishes a useful coordinate transformation for the error system of an emulation-based controller.

Lemma 4. *Consider the process and the (open-loop) state estimator*

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(t_k) = x_0 \quad \forall t \in [t_k, t_{k+1}) \tag{46}$$

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t), \quad \hat{x}(t_k) = \hat{x}_0 \quad \forall t \in [t_k, t_{k+1}). \tag{47}$$

There exists a time-varying matrix $P(t) \in \mathbb{R}^{n\times n}$ such that for any $t_k, t_{k+1}, x_0, \hat{x}_0$, the state estimation error

$$e(t) \coloneqq P(t)(x(t) - \hat{x}(t)) \tag{48}$$

satisfies

$$e_i(t) = e^{a_i(t-t_k)} G_i(t-t_k) e_i(t_k), \quad e_i(t) \in \mathbb{R}^{d_i}, \tag{49}$$

for all $t \in [t_k, t_{k+1})$ *and all* $i \in \{1, \ldots, n_b\}$*, where* n_b *is the number of real Jordan blocks in the real Jordan normal form of* A*,* a_i *is the real part of the eigenvalue associated with Jordan block* i*, and* d_i *is the geometric multiplicity of that eigenvalue; the time-varying real matrix* $G_i(t)$ *has the form*

$$G_i(t) \coloneqq \begin{bmatrix} 1 & t & \frac{t^2}{2!} & \cdots & \frac{t^{d_i-1}}{(d_i-1)!} \\ & 1 & t & & \\ & & \ddots & & \\ & & & & \\ & & & & 1 \end{bmatrix} \in \mathbb{R}^{d_i \times d_i} \tag{50}$$

if the i*th Jordan block corresponds to a real eigenvalue, and*

$$G_i(t) \coloneqq \begin{bmatrix} I_2 & I_2 t & I_2 \frac{t^2}{2!} & \cdots & I_2 \frac{t^{d_i-1}}{(d_i-1)!} \\ & I_2 & I_2 t & & \\ & & \ddots & & \\ & & & & \\ & & & & I_2 \end{bmatrix} \in \mathbb{R}^{2d_i \times 2d_i} \tag{51}$$

if it corresponds to a complex conjugate pair, where $I_2 \coloneqq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$*. Moreover, there exists a positive scalar* ϵ_P *for which*

$$\sigma_{\min}(P(t)) \geqslant \epsilon_P \qquad \forall t \geqslant 0, \tag{52}$$

where $\sigma_{\min}(\cdot)$ *denotes the smallest singular value of a matrix.*

4.1 Proof of Theorem 2

The basic idea of the proof is as follows. The encoder and decoder each run internal copies of the process to compute an estimate $\hat{x}$ of the state. Since there is no channel noise, the encoder's and decoder's state estimates will be equal, which corresponds to an information pattern "encoder class 1a" in the terminology of [22].

The encoder monitors the state estimation error and periodically transmits symbols to the decoder that essentially encode a quantized version of the error, making sure that the average cost per symbol does not exceed γ. The decoder then uses those symbols to update its state estimate $\hat{x}$.

4.1.1 Definition of the encoding and decoding scheme

We first select the integers M and N for our M-of-N encoder. Assume that S, r, and γ satisfy (45), so that

$$\eta \coloneqq rf(\gamma, S)\ln 2 - \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A] > 0. \tag{53}$$

In view of (10) and (11), we conclude that we can pick N sufficiently large to satisfy

$$rf(\gamma, S)\ln 2 - r\,\frac{\ln L(N, N\gamma, S)}{N\ln(S+1)}\ln 2 < \eta/2, \tag{54}$$

and we then define $M \coloneqq N\gamma$. By Lemma 5 in the appendix, this encoder has an average cost per symbol not exceeding γ.

Now we specify which N-length codewords will be transmitted. Here is the basic idea: The encoder and decoder each estimate the state of the process as $\hat{x}(t)$ as defined in (47), with $t_0 \coloneqq 0$ and $\hat{x}(t_0) \coloneqq 0$. The encoder monitors the state estimation error $e(t) \coloneqq P(t)(x(t) - \hat{x}(t))$, where $P(t)$ is determined by Lemma 4. For each of the n_b error subsystems $e_i(t) \in \mathbb{R}^{d_i}$ given by (49) we employ a *sub-encoder* i that monitors $e_i(t)$ and every T_i time units (to be defined shortly) transmits to the decoder a set of N-length codewords with M or fewer non-free symbols from the alphabet $\{0, \ldots, S\}$. The chosen set of codewords is essentially the index of the d_i-dimensional quantization cell in which $e_i(kT_i) \in \mathbb{R}^{d_i}$ lies. Based on this set of codewords, the encoder and decoder each adjusts their state estimates, and the procedure repeats.

We now define the scheme formally. We first select the transmission periods T_i: partition the n_b error systems based on whether or not they are stable:

$$\mathcal{S} \coloneqq \{i \in \{1, \ldots, n_b\} : a_i < 0\}$$
$$\mathcal{U} \coloneqq \{i \in \{1, \ldots, n_b\} : a_i \geqslant 0\},$$

where a_i is the real part of the ith eigenvalue of A. For the subsequent argument, in the case that $a_i = 0$ we add a small positive number to it so that (45) still holds, and use the same label a_i to denote this number. Note that, in contrast with the previous section, we treat eigenvalues with zero real part as unstable.

The error dynamics for e_i with $i \in \mathcal{S}$ are stable and so there is no need to transmit information on behalf of e_i, $i \in \mathcal{S}$, since these errors will converge to zero exponentially fast. So there is no need to define T_i for $i \in \mathcal{S}$. For $i \in \mathcal{U}$, we select the transmission period for sub-encoder i to be

$$T_i \coloneqq c_i \frac{\ln L(N, M, S)}{a_i} \frac{1}{1 + \eta/(2\sum_{i:\Re\lambda_i[A]>0}\lambda_i[A])}, \tag{55}$$

where the positive integer c_i is chosen large enough so that T_i satisfies

$$\sum_{j=0}^{d_j-1} \frac{T_i^j}{j!} < e^{\kappa T_i} \tag{56}$$

where $\kappa \coloneqq a_i\eta/(4\sum_{i:\Re\lambda_i[A]>0}\lambda_i[A]) > 0$.

Note that for those eigenvalues whose real part was 0, the transmission period can be arbitrarily large (but finite) because the positive number that was added to them can be arbitrarily small.

Now we specify how the sub-encoder i selects which codeword to transmit. For $i \in \mathcal{S}$ no symbols are transmitted. For $i \in \mathcal{U}$, the ith sub-encoder initializes with $L_{i,0} \coloneqq \sup_{x_0\in\mathcal{X}_0}\|x_0\|_\infty$ and at time kT_i, $k \in \mathbb{N}_{>0}$, performs the following steps:

1. Divide the d_i-dimensional box $e^{(a_i+\kappa)T_i}L_{i,k-1}[-1,1]^{d_i}$ into $L(N,M,S)^{c_id_i}$ smaller boxes of equal size by dividing each of its d_i dimensions into $L(N,M,S)^{c_i}$ intervals of equal length. the sub-encoder i determines in which of these boxes the error $e_i(kT_i)^-$ lies and transmits this information to the decoder. Since there are $L(N,M,S)^{c_id_i}$ boxes, this requires sending exactly c_id_i M-of-N codewords.

   Let $B_{i,k} \subset \mathbb{R}^{d_i}$ denote the indicated box, $b_{i,k} \in \mathbb{R}^{d_i}$ denote the box's center, and $w_{i,k}$ denote the transmitted set of codewords. Note that set $B_{i,k} - b_{i,k} \subset \mathbb{R}^{d_i}$ is a cube centered at 0.

2. Update the state estimate as

$$\hat{x}(kT_i)^+ = \hat{x}(kT_i)^- + I_i'b_{i,k}, \tag{57}$$

   where $\hat{x}^+(t) \coloneqq \lim_{\tau\downarrow t}\hat{x}(\tau)$ and $\hat{x}^-(t) \coloneqq \lim_{\tau\uparrow t}\hat{x}(\tau)$, and the matrix $I_i \in \mathbb{R}^{d_i\times n}$ "extracts" from the error $e(t)$ its component $e_i(t)$ such that $e_i(t) = I_ie(t)$. Specifically,

$$I_i \coloneqq \begin{bmatrix} 0_{d_i\times d_1} & 0_{d_i\times d_2} & \dots & I_{d_i\times d_i} & \dots & 0_{d_i\times d_{n_b}} \end{bmatrix}.$$

3. Define

$$L_{i,k} \coloneqq \sup_{z\in B_{i,k}-b_{i,k}}\|z\|_\infty \tag{58}$$

The sequences $\{w_{i,k}\}$, $\{B_{i,k}\}$, $\{b_{i,k}\}$, and $\{L_{i,k}\}$ are available both to the encoder and the decoder, so the decoder can maintain and update its own state estimate via Step 2, which is used by the state feedback controller $u \coloneqq K\hat{x}$. We now show that the proposed encoding/decoding scheme satisfies the conditions of Theorem 2, namely that the state goes to 0 and that the average bit-rate is at most r.

4.1.2 The scheme exponentially stabilizes the process

From process (1) and the definition of $e(t)$ in (48), the control law $u = K\hat{x}$ results in the following closed-loop dynamics:

$$\dot{x}(t) = (A+BK)x(t) - BKe(t). \tag{59}$$

Since $A + BK$ is Hurwitz, the state $x(t)$ converges exponentially to 0 provided that $e(t) \to 0$ exponentially. We now prove that $e(t) \to 0$ exponentially under the proposed scheme.

The basic idea is as follows: On one hand, in view of (49) and (56), the error $e_i(t)$ grows in magnitude by a factor less than $e^{(a_i+\kappa)T_i}$ in the T_i time units between the transmission of sets of codewords. On the other hand, every T_i time units the ith sub-encoder sends $L(N, M, S)^{c_i d_i}$ codewords, allowing the ith sub-decoder to reduce its uncertainty of $e_i(t)$ by a factor of $L(N, M, S)^{c_i d_i}$. We will show that condition (45) in Theorem 2 implies that $L(N, M, S)^{c_i d_i} > e^{(a_i+\kappa)T_i}$, meaning that the sub-decoder's uncertainty in $e_i(t)$ shrinks faster than the error dynamics expands $e_i(t)$. Therefore the decoder can determine $e(t)$ and drive it to 0.

First we prove by induction that the rule (57) for updating the state estimate guarantees that $\|e_i(kT_i)^+\|_\infty \leqslant L_{i,k}$. From the definition of $e(t)$ and I_i we have

$$e_i(kT_i)^- = I_i e(kT_i)^- = I_i \left(x(kT_i)^- - \hat{x}(kT_i)^-\right). \tag{60}$$

Solving the update rule (57) for $\hat{x}(kT_i)^-$ and substituting the result into (60) yields

$$\begin{aligned} e_i(kT_i)^- &= I_i \left(x(kT_i)^- - \left(\hat{x}(kT_i)^+ - I_i' b_{i,k}\right)\right) \\ &= e_i(kT_i)^+ + b_{i,k}, \end{aligned} \tag{61}$$

where we used the fact that $x(kT_i)^- = x(kT_i)^+$ due to the continuity of the solution $x(t)$. Next, suppose by the induction hypothesis that $\|e_i((k-1)T_i)^+\|_\infty \leqslant L_{i,k-1}$. Then we have

$$\|e_i((k-1)T_i)^+\|_\infty \leqslant L_{i,k-1} \tag{62}$$

$$\Leftrightarrow \quad e_i((k-1)T_i)^+ \in L_{i,k-1}[-1,1]^{d_i} \tag{63}$$

$$\Rightarrow \quad e_i(kT_i)^- \in e^{a_i T_i}\|G_i(T_i)\|_\infty L_{i,k-1}[-1,1]^{d_i} \tag{64}$$

$$\Rightarrow \quad e_i(kT_i)^- \in e^{(a_i+\kappa)T_i} L_{i,k-1}[-1,1]^{d_i}, \tag{65}$$

where (64) holds because $e_i(t)$ follows the dynamics (49) between transmissions, and (65) follows because T_i was chosen to satisfy (56) and we have $\|G_i(T_i)\|_\infty = \sum_{j=0}^{d_j-1} T_i^j/j!$.

Moreover, the set in (65) is precisely the box in Step 1 of the proposed scheme, so therefore we must have $e_i(kT_i)^- \in B_{i,k}$. Applying (61) yields $e_i(kT_i)^+ \in B_{i,k} - b_{i,k}$, and therefore

$$\|e_i(kT_i)^+\|_\infty \leqslant \sup_{z \in B_{i,k} - b_{i,k}} \|z\|_\infty =: L_{i,k}. \tag{66}$$

This demonstrates that $\|e_i(kT_i)^+\|_\infty \leqslant L_{i,k}$ for all $k \in \mathbb{N}_{>0}$.

From Step 1 of the encoding scheme, the length $L_{i,k}$ is essentially the side-length of the cube $B_{i,k}$. The set $B_{i,k}$ was constructed by dividing every dimension of $e^{(a_i+\kappa)T_i} L_{i,k-1}[-1,1]^{d_i}$ into $L(N, M, S)^{c_i}$ pieces. Therefore the lengths $L_{i,k}$ are recursively related via

$$L_{i,k} = \frac{e^{(a_i+\kappa)T_i}}{L(N, M, S)^{c_i}} L_{i,k-1}, \tag{67}$$

and therefore

$$L_{i,k} = e^{Rk} L_{i,0}, \tag{68}$$

where

$$R \coloneqq \ln\left(\frac{e^{(a_i+\kappa)T_i}}{L(N,M,S)^{c_i}}\right). \tag{69}$$

The transmission period T_i and κ were chosen in (55) to satisfy

$$\frac{e^{(a_i+\kappa)T_i}}{L(N,M,S)^{c_i}} < 1, \tag{70}$$

and so $R < 0$. Therefore the event boundaries $L_{i,k}$ shrink to 0 at an exponential rate.

This implies that $e_i(t) \to 0$ exponentially, as follows. For any time t we have $t = kT_i + \underline{t}$, where $k \coloneqq \lfloor t/T_i \rfloor$ and $\underline{t} \in [0, T_i)$. Therefore

$$\begin{aligned}
\|e_i(t)\|_\infty &= \|e_i(kT_i + \underline{t})\|_\infty && (71)\\
&\leqslant e^{a_i \underline{t}} \|G_i(\underline{t})\|_\infty \|e_i(kT_i)^+\|_\infty && (72)\\
&\leqslant e^{a_i T_i} \|G_i(T_i)\|_\infty \|e_i(kT_i)^+\|_\infty && (73)\\
&\leqslant e^{a_i T_i} \|G_i(T_i)\|_\infty L_{i,k} && (74)\\
&= e^{a_i T_i} \|G_i(T_i)\|_\infty L_{i,0} e^{Rk} && (75)\\
&\leqslant e^{a_i T_i} \|G_i(T_i)\|_\infty L_{i,0} e^{-R} e^{Rt/T_i}, && (76)
\end{aligned}$$

where (72) follows from the error dynamics (49), (74) follows from (66), and (75) follows from (68). Since $R < 0$, this establishes that $e_i(t) \to 0$ at an exponential rate. Since this holds for all i, $e(t)$ exponentially converges to 0 as well. Therefore by (59), the state $x(t)$ exponentially converges to 0.

4.1.3 The scheme's average bit-rate does not exceed r

Since each sub-encoder is transmitting independently, the average bit-rate of this encoding scheme as a whole is simply the sum of the sub-encoder's average bit-rates. For $i \in \mathcal{S}$, the ith sub-encoder never transmits. For $i \in \mathcal{U}$, every T_i time units the ith sub-encoder sends $c_i d_i$ codewords, each from a codeword library of length $L(N,M,S)$. Therefore its average bit-rate is $r_i \coloneqq c_i d_i \log_2 L(N,M,S)/T_i$. The encoder's total average bit-rate is therefore

$$\sum_{i\in\mathcal{U}} r_i = \log_2 L(N,M,S) \sum_{i\in\mathcal{U}} \frac{c_i d_i}{T_i}.$$

Leveraging (55) yields

$$\sum_{i\in\mathcal{U}} r_i \leqslant \frac{1}{\ln 2}\left(1 + \frac{\eta}{2\sum_{i:\Re\lambda_i[A]>0}\lambda_i[A]}\right)\sum_{i\in\mathcal{U}} d_i a_i. \tag{77}$$

Since $\mathcal{U}$ contains the non-negative real parts of the eigenvalues of A, we have $\sum_{i\in\mathcal{U}} d_i a_i = \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A]$. From this and (77) we conclude that

$$\begin{aligned}\sum_{i\in\mathcal{U}} r_i &\leqslant \frac{1}{\ln 2}\left(\sum_{i:\Re\lambda_i[A]>0} \lambda_i[A] + \frac{\eta}{2}\right)\\ &< r\frac{\ln L(N,M,S)}{N\ln(S+1)} \leqslant r,\end{aligned} \tag{78}$$

where in (78) we leveraged (53) and (54) and then used the fact that L is nonincreasing in its second argument and so $L(N,M,S) \leqslant L(N,N,S) = (S+1)^N$. We conclude that this encoding scheme has average bit-rate less than r.

This concludes the proof of Theorem 2. ∎

An unexpected consequence of Theorems 1 and 2 is that when it is possible to drive the state of process (1) to 0 with a given average bit-rate r, one can always find M-of-N encoders that stabilize it for (essentially) the same average bit-rate and average cost per symbol not exceeding $S/(S+1)$, i.e., approximately a fraction $1/(S+1)$ of the symbols will not consume communication resources. In the most advantageous case, the encoder/decoder use the alphabet $\{0,1\}$ and the encoder's symbol stream consumes no more than 50% of the communication resources.

The following summarizes this observation.

Corollary 1. *If process (1) can be bounded with an encoder/decoder pair with average bit-rate r, then for any $\epsilon > 0$ and $S \in \mathbb{N}_{>0}$ there exists an M-of-N encoder using alphabet $\{0,\ldots,S\}$ with average bit-rate $r+\epsilon$ and average cost per symbol not exceeding $S/(S+1)$ that bounds its state.*

Proof of Corollary 1. Since the original encoder/decoder pair bounds the state, then by (4) we have

$$\begin{aligned}\sum_{i:\Re\lambda_i[A]>0} \lambda_i[A] &\leqslant r\ln 2 < (r+\epsilon)\ln 2\\ &= (r+\epsilon) f\left(\frac{S}{S+1}, S\right)\ln 2.\end{aligned}$$

Applying Theorem 2 completes the proof. ∎

The price paid for using an encoder/decoder with average cost per symbol close to $S/(S+1)$ is that it may require prohibitively long codewords (large N) as compared to an encoder with higher average cost per symbol. To see this, note that $f(\gamma,S) = 1$ when $\gamma \in [S/(S+1),1]$ and recall that $\ln L(N,N\gamma,S)/N$ is monotonically nondecreasing in γ and N. Hence, with r and S fixed, one can decrease γ from 1 toward $S/(S+1)$ and still satisfy (54) by increasing N. This can be seen in Figure 3.

Remark 3. In the problem statement, $x(0)$ was assumed to belong to a known bounded set. If the region $\mathcal{X}_0$ is not precisely known, the proposed scheme could be modified by introducing an initial "zooming-out" stage as described in [3], where the encoder picks an arbitrary box to quantize and successively zooms out at a super-linear rate until the box captures the state.

5 Event-driven encoders

In Section 4 we constructed an N-of-M encoding scheme that stabilizes process (1) provided that the bit-rate and average cost condition (45) holds. This scheme may be difficult to implement in practice if the encoder/decoder pair use a large number of codewords. In this section we present an *event-based* encoding scheme that is easy to implement and does not require storing a large set of codewords. Instead, it uses a library of only three symbols $\{-1, 0, 1\}$ and does not group them into codewords. The basic idea is to monitor in parallel each one-dimensional component of the error system, and as long as it stays inside a fixed interval, send the free symbol 0. A non-free symbol is sent only when the one-dimensional component of the error leaves the interval: send -1 if the error exited the left side of the interval and send 1 if it exited out the right side. Communication resources are therefore consumed only upon the occurrence of this event, justifying the label *event-based*. The proposed scheme resembles the distributed-sensor scheme of [22], in that each coordinate of a plant measurement is sent by a dedicated encoder to a central decoder.

The proposed scheme has similarities with the one from Section 4 in the following ways: the encoder and decoder each estimate the process as $\hat{x}$ using (47); the emulation-based controller is $u \coloneqq K\hat{x}$, where K is a stabilizing state-feedback gain; Lemma 4 decouples the error system into n_b sub-systems; each of n_b sub-encoders monitors the d_i-dimensional component of the error and transmits a block of symbols every T_i time units; only the unstable systems $\mathcal{U}$ require transmission. If A is diagonalizable over $\mathbb{C}$, then this event-based encoding scheme reduces to the one proposed in [17].

Unlike the scheme from Section 4, this scheme differs in what symbols are sent and how the state estimate $\hat{x}$ is updated: For $i \in \mathcal{U}$, at time kT_i, $k \in \mathbb{N}_{>0}$ (with T_i to be determined shortly), the sub-encoder i monitors the d_i scalar components $e_{i,j}(t) \in \mathbb{R}$, $j \in \{1, \dots, d_i\}$ of $e_i(t)$, and for each one sends a symbol $s_{i,j}(k) \in \{-1, 0, 1\}$ according to

$$s_{i,j}(k) = \begin{cases} -1 & e_{i,j}(kT_i) < -L_j \\ 0 & e_{i,j}(kT_i) \in [-L_j, L_j] \\ 1 & e_{i,j}(kT_i) > L_j \end{cases} \qquad k \in \mathbb{N}_{>0}, \tag{79}$$

with the *event boundaries* $L_j > 0$ also to be determined shortly. The encoder and decoder then each update their state estimates as

$$\hat{x}(kT_i)^+ = \hat{x}(kT_i)^- + P(kT_i)^{-1}\mathbf{v}_{i,j}\Delta_{i,j}(s_{i,j}(k)), \\ i \in \{1, \dots n\},\ k \in \mathbb{N}_{>0}, \tag{80}$$

where the unit vector $\mathbf{v}_{i,j} \in \mathbb{R}^{d_i}$ satisfies $e_{i,j}(t) = \mathbf{v}'_{i,j} e(t)$, $\hat{x}(t)^+$ and $\hat{x}(t)^-$ denote limiting values of $\hat{x}(t)$ from above and below t, $P(t)$ is from Lemma 4, and the decoding function $\Delta_{i,j} : \{-1, 0, 1\} \to \mathbb{R}$ is defined as

$$\Delta_{i,j}(s) \coloneqq \begin{cases} -\frac{L_j}{2}(1 + \exp(a_i T_i)) & s = -1 \\ 0 & s = 0 \\ \frac{L_j}{2}(1 + \exp(a_i T_i)) & s = 1, \end{cases} \tag{81}$$

where $a_i \coloneqq \Re\lambda_i[A]$ is defined as before. Note that the nonzero values of $\Delta_{i,j}$ are merely the midpoints of the intervals $[L_j, L_j \exp(a_i T_i)]$ and $[-L_j, -L_j \exp(a_i T_i)]$.

The event-based encoding/decoding scheme and controller are described in pseudo-code as Algorithms 1 and 2 below.

Algorithm 1. *(Encoder)*

Set state estimate $\hat{x}(0) \leftarrow 0$
Continuously compute state estimate $\hat{x}(t)$ *from* (47)
for *each sub-encoder* $i \in \mathcal{U}$ *in parallel,* ***do***
  for *time* $t = kT_i$, $k \in \{1, 2, \ldots\}$ ***do***
    measure state $x(t)$ *and compute* $e_i(t)$ *from* (48)
    for *each scalar component* $e_{i,j}(t)$, $j \in \{1, \ldots, d_i\}$, ***do***
      compute $s_{i,j}(k)$ *from* (79) *and transmit it to decoder*
      update $\hat{x}(t)$ *from* (80)
    end for
  end for
end for

Algorithm 2. *(Decoder)*

Set state estimate $\hat{x}(0) \leftarrow 0$
Continuously compute state estimate $\hat{x}(t)$ *from* (47)
Continuously compute actuation signal $u(t) \coloneqq K\hat{x}(t)$
for *each sub-decoder* $i = 1$ *to* n *in parallel,* ***do***
  for *time* $t = kT_i$, $k = 1, 2, \ldots$ ***do***
    receive $s_{i,j}(k)$ *from the encoder*
    update $\hat{x}(t)$ *from* (80)
  end for
end for

This concludes the description of the event-based encoder/decoder pair, except for the precise choice of the transmission periods T_i and the event boundaries L_j. The following result states that if the average bit-rate and average cost per symbol satisfy a particular condition, then one can choose transmission periods and event boundaries for which this scheme obeys the communication constraints and bounds the process state.

Theorem 3. *Consider process (1), and assume that $A + BK$ is Hurwitz. For every $\gamma \in [0, 1]$ and $r > 0$ satisfying*

$$r \frac{h^{-1}(\gamma)}{\ln 3} \ln 2 \geqslant \sum_{i:\Re\lambda_i[A]>0} \lambda_i[A], \tag{82}$$

$$h(x) \coloneqq \frac{x}{\ln \frac{2}{e^x - 1}}, \ x \in (0, \ln 3), \ h(0) \coloneqq 0, \tag{83}$$

there exists an emulation-based controller and event-based encoder/decoder pair of the type described above that keeps the state of the process bounded for every initial condition in $\mathcal{X}_0$; the encoder has average bit-rate not exceeding r and has average cost per symbol not exceeding γ.

Remark 4. Whereas the necessary and sufficient bounds from Theorems 1 and 2 had the term $f(\gamma, S)$, the event-based encoding bound in (82) has the term $h^{-1}(\gamma)/\ln 3$. The ratio $g(\gamma, S) \coloneqq f(\gamma, S)/(h^{-1}(\gamma)/\ln 3)$ captures the factor by which the event-based bound exceeds the tight theoretical bound developed in the previous sections. This factor is a function of the encoder's average cost per symbol γ and the alphabet size S, and is plotted in Figure 4 for $S = 2$ and $S = 1$. Since the event-based encoder has $S = 2$, the $g(\gamma, 2)$ curve provides a "fair" comparison between the event-based encoder and all other encoders with alphabet size $S = 2$. The $g(\gamma, 1)$ curve compares the event-based encoder with all other encoders with the smallest (most efficient) alphabet, $S = 1$. We observe:

- $g(\gamma, 1) < 2.43$ for all $\gamma \in (0, 1]$.
- $g(\gamma, 2) < 2.0$ for all $\gamma \in (0, 1]$.
- $g(1, S) = \ln 3/\ln 2 \approx 1.58$ for all $S \in \mathbb{N}_{>0}$.

The first point guarantees that this encoding and control scheme is never more than 2.43 times more conservative than the optimal bound established in Theorems 1 and 2. Specifically, if a given process may be bounded with a certain average bit-rate r, then there exists an average bit-rate $\widetilde{r}$ not exceeding $2.43r$ such that this event-based scheme can bound the process using average bit-rate $\widetilde{r}$. The second point establishes that this event-based scheme never requires more than twice the average bit-rate of any stabilizing N-of-M encoding scheme that, like this scheme, uses a three-symbol alphabet. The third point states that as the communication constraint relaxes ($\gamma \to 1$), this event-based encoding scheme is only 1.58 times more conservative than the optimal average bit-rate bound from Theorems 1 and 2. A consequence of $g(\gamma, S) > 1$ is that event-based encoders are sub-optimal in the following sense: if r, γ, and S satisfy (82), then there exists $\widetilde{r} \coloneqq r/g(\gamma, S) < r$ for which $\widetilde{r}$, γ, and S satisfy (45). Therefore, whenever Theorem 3 could be invoked with (r, γ, S) to build a stabilizing event-based encoding scheme, one could instead invoke Theorem 2 with $(\widetilde{r}, \gamma, S)$ to construct a stabilizing

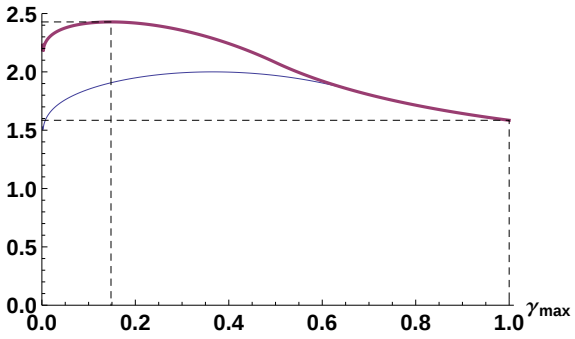

Figure 4: Plot of $g(\gamma, S)$ versus γ, for $S = 1$ (thick solid line) and $S = 2$ (thin solid line).

M-of-N encoding scheme with a smaller average bit-rate. This is the price paid for the convenience of the simple event-based logic as opposed to having to implement an encoder/decoder with a (possibly quite large) library of M-of-N codewords.

Remark 5. In Remark 2 it was noted that the sufficiency result in Theorem 2 would not bound the process state if the data-rate condition (45) held only with equality. In contrast, if the present data-rate inequality (82) holds with equality, the following event-based scheme bounds the state of the process, as we will show in the proof of Theorem 3. However, the two sufficiency results of Theorem 2 and Theorem 3 are consistent in the sense that if their data-rate conditions [(45) and (82) respectively] hold with strict inequality, then exponential stabilization can be achieved, with the rate of exponential convergence determined by the "gap" in the inequality. To see this for the present scheme, suppose (82) holds with strict equality and let $\overline{x}(t) \coloneqq e^{\epsilon t}x(t)$, where $\epsilon > 0$ is small enough that

$$r\frac{h^{-1}(\gamma)}{\ln 3}\ln 2 > \sum_{i:\Re\lambda_i[A]>0}\lambda_i[A] + n\epsilon, \tag{84}$$

and suppose $A + \epsilon I + BK$ is Hurwitz. Applying Theorem 3 to the $\overline{x}$ system provides a controller and encoder/decoder that bounds $\overline{x}$. However, $\|\overline{x}(t)\| \leqslant c$ is equivalent to $\|x(t)\| < ce^{-\epsilon t}$, so the state $x(t)$ converges to 0 exponentially fast.

5.1 Proof of Theorem 3

The main idea behind the proof is to show that, when assumption (82) holds, it is possible to allocate the available average bit-rate among sub-encoders in such a way that each sub-encoder has a sufficiently large average bit-rate to bound its components of the state estimation error.

For the sub-encoder $i \in \mathcal{U}$, we pick the transmission period T_i as

$$T_i \coloneqq h^{-1}(\gamma)/(a_i + \eta), \tag{85}$$

where the definition of h is from (83) and $\eta > 0$ satisfies

$$r\frac{h^{-1}(\gamma)}{\ln 3}\ln 2 \geqslant \sum_{i:\Re\lambda_i[A]>0}\lambda_i[A] + n\eta. \tag{86}$$

As mentioned above, no information needs to be sent on behalf of the stable systems $i \in \mathcal{S}$.

The event boundaries $L_j > 0$ are chosen as follows. Define

$$\underline{\tau}_i \coloneqq \frac{1}{a_i}\ln\left(\frac{2}{e^{(a_i+\eta)T_i}-1}\right). \tag{87}$$

Note that $\infty > \underline{\tau}_i > 0$ because $a_i > 0$ for $i \in \mathcal{U}$ and $\frac{2}{e^{(a_i+\eta)T_i}-1} > 1$ by our choice of T_i. Next, pick $1 > \phi > 0$ sufficiently small so that

$$\phi < e^{-T_i}/4 \tag{88}$$

$$\underline{\tau}_i \leqslant \frac{1}{a_i}\ln\left(\frac{2}{(e^{a_i T_i}(1+2e^{T_i}\phi)-1+2e^{\underline{\tau}_i}\phi}\right) \tag{89}$$

for all $i \in \mathcal{U}$. Finally, define the event boundaries recursively as

$$L_n \coloneqq \sup_{x_0\in\mathcal{X}_0}\|P(0)x_0\|_\infty \tag{90}$$

$$L_j \coloneqq \frac{1}{\phi}\sum_{l=j+1}^{n} L_l \quad j \in \{1,\ldots,n-1\}. \tag{91}$$

5.1.1 The scheme's average bit-rate does not exceed r

For $i \in \mathcal{U}$, sub-encoder i sends d_i symbols from the alphabet $\{-1,0,1\}$ every T_i time units, resulting in an average bit-rate of

$$r_i \coloneqq d_i \log_2 3/T_i, \tag{92}$$

and so the average bit-rate used by the encoder as a whole is simply

$$\sum_{i\in\mathcal{U}} r_i = \log_2 3\sum_{i\in\mathcal{U}}\frac{d_i}{T_i} = \frac{\log_2 3}{h^{-1}(\gamma)}\sum_{i\in\mathcal{U}} d_i(a_i+\eta) \tag{93}$$

$$= \frac{\log_2 3}{h^{-1}(\gamma)}\left(\sum_{i:\Re\lambda_i[A]>0}\lambda_i[A] + n\eta\right) \leqslant r, \tag{94}$$

where the last inequality follows from hypothesis (82). Hence, this encoding scheme uses an average bit-rate of r or less.

5.1.2 The scheme stabilizes the process

Next we show that this controller and event-based encoder/decoder pair bound the state of process (1). In view of (59), this is ensured if $e(t)$ is bounded. Since $e_i(t) \to 0$ for $i \in \mathcal{S}$, we focus on $e_i(t)$ for $i \in \mathcal{U}$.

We proceed with an inductive proof that the sequence $\{e_{i,j}(kT_i)^+\}_{k\in\mathbb{N}_{>0}}$ is bounded for $i \in \mathcal{U}$, $j \in \{1, \dots, d_i\}$. The base of induction $k = 0$ follows from the definition of L_j in (90). Next we prove that $e_{i,j}(kT_i)^+ \in [-L_j, L_j]$ provided that $e_{i,l}(kT_i - T_i)^+ \in [-L_l, L_l]$ for $l \in \{j, \dots, d_i\}$. If $e_{i,j}(kT_i - T_i)^+$ is so small that it does not grow outside the box $[-L_j, L_j]$ by the next timestep, then we naturally have $e_{i,j}(kT_i)^+ \in [-L_j, L_j]$. On the other hand, suppose at a specific time t^* satisfying $kT_i - T_i \leqslant t^* < kT_i$, the scalar error $e_{i,j}(t^*)$ grows to the boundary of the box $[-L_j, L_j]$; without loss of generality suppose $e_{i,j}(t^*) = L_j$. Up to T_i time units later, the timestep kT_i occurs and the sub-encoder i transmits $s_{i,j}(k) = 1$ to the decoder. Upon receiving symbol 1, the decoder knows from the encoding scheme (79) that the scalar error $e_{i,j}(kT_i)^-$ immediately before the transmission must have exceeded the event boundary L_j and hence $e_{i,j}(kT_k)^- > L_j$. Moreover,

$$
\begin{aligned}
|e_{i,j}(kT_i)^-| &= |\mathbf{v}'_{i,j} e_i(kT_i)^-| && (95)\\
&= |\mathbf{v}'_{i,j} e^{a_i T_i} G_i(T_i) e_i(kT_i - T_i)^+| && (96)\\
&\leqslant e^{a_i T_i} \left| \sum_{l=0}^{d_i - j} \frac{T_i^l}{l!} e_{i,j+l}(kT_i - T_i)^+ \right| && (97)\\
&\leqslant e^{a_i T_i} \Bigg(|e_{i,j}(kT_i - T_i)^+| \\
&\qquad + \sum_{l=1}^{d_i - j} \frac{T_i^l}{l!} \sum_{l=1}^{d_i - j} |e_{i,j+l}(kT_i - T_i)|^+ \Bigg) && (98)\\
&\leqslant e^{a_i T_i} \left(L_j + \sum_{l=1}^{d_i - j} \frac{T_i^l}{l!} \sum_{l=1}^{d_i - j} L_{j+l} \right) && (99)\\
&\leqslant e^{a_i T_i} L_j \left(1 + e^{T_i} \phi\right), && (100)
\end{aligned}
$$

where $\mathbf{v}_{i,j} \in \mathbb{R}^{d_i}$ is a unit vector satisfying (95), (96) follows from the error dynamics (49) in Lemma 4, (97) follows from the definition of the matrix $G_i(T_i)$, (98) follows from the triangle inequality, (99) follows from the induction hypothesis, and (100) follows by the definition of ϕ, and by upper-bounding the sum $\sum_{l=1}^{d_i - j} T_i^l / l!$ by e^{T_i}. Therefore the decoder can conclude that

$$
e_{i,j}(kT_i)^- \in (L_j, L_j\, e^{a_i\, T_i}(1 + e^{T_i}\phi)]. \tag{101}
$$

We can express the scalar error $e_{i,j}(kT_i)^-$ as the overall error vector $e(kT_i)^- \in \mathbb{R}^n$ times an appropriate unit vector:

$$
e_{i,j}(kT_i)^- = \mathbf{v}'_{i,j} e(kT_i)^- \tag{102}
$$

$$= \mathbf{v}'_{i,j} P(kT_i)(x(kT_i)^- - \hat{x}(kT_i)^-). \tag{103}$$

Rearranging the update rule (80) yields an expression for $\hat{x}(kT_i)^-$:

$$\hat{x}(kT_i)^- = \hat{x}(kT_i)^+ - P(kT_i)^{-1}\mathbf{v}_{i,j}\Delta_{i,j}(1). \tag{104}$$

Substituting this into (103) yields

$$\begin{aligned} e_{i,j}(kT_i)^- &= \mathbf{v}'_{i,j} P(kT_i)\left(x(kT_i)^- - \right. \\ &\qquad \left. \hat{x}(kT_i)^+ + P(kT_i)^{-1}\mathbf{v}_{i,j}\Delta_{i,j}(1)\right) \\ &= \mathbf{v}'_{i,j} P(kT_i)\left(x(kT_i)^- - \hat{x}(kT_i)^+\right) + \Delta_{i,j}(1) \\ &= e_{i,j}(kT_i)^+ + \Delta_{i,j}(1), \end{aligned}$$

where we used the fact that $x(kT_i)^- = x(kT_i)^+$ due to the continuity of the solution $x(t)$. Substituting this into (101) yields

$$e_{i,j}(kT_i)^+ + \Delta_{i,j}(1) \in (L_j, L_j\, e^{a_i\, T_i}(1 + e^{T_i}\phi)] \tag{105}$$

which is equivalent to

$$e_{i,j}(kT_i)^+ \in \left(-\frac{L_j(e^{a_i\, T_i} - 1)}{2}, \frac{L_j(e^{a_i\, T_i}(1 + 2e^{T_i}\phi) - 1)}{2}\right]. \tag{106}$$

Recall that T_i was chosen to satisfy $h(a_i\, T_i) = \gamma \leqslant 1$. Applying h^{-1} to this yields $a_i\, T_i \leqslant \ln 2$, and so $e^{a_i\, T_i} \leqslant 2$. Combining this with the upper bound (88) on ϕ yields

$$\frac{L_j(e^{a_i\, T_i}(1 + 2e^{T_i}\phi) - 1)}{2} < L_j. \tag{107}$$

Applying this to (106) establishes that

$$e_{i,j}(kT_i)^+ \in (-L_j, L_j) \tag{108}$$

and completes the inductive proof that the sequence $\{e_{i,j}(kT_i)^+\}_{k\in\mathbb{N}_{>0}}$ is bounded. Since this holds for arbitrary $j \in \{1, \ldots, d_i\}$, the sequence $\{e_i(kT_i)^+\}_{k\in\mathbb{N}_{>0}} \subset \mathbb{R}^{d_i}$ is also bounded. Following a similar argument to (71), we conclude that $e_i(t)$ is bounded for any $t \geqslant 0$. Since $e_i(t)$ is bounded for all $i \in \mathcal{U}$ and $e_j(t) \to 0$ for $j \in \mathcal{S}$, this controller and encoder/decoder pair bound the estimation error. Therefore the state is bounded for all time as well.

5.1.3 The scheme's average cost per symbol does not exceed γ

Lastly we prove that this encoding scheme has average cost per symbol not exceeding γ. The symbol stream emitted by the encoder is comprised of the $|\mathcal{U}|$ individual symbol sequences $\{s_{i,j}(k)\}_{k\in\mathbb{N}_{>0}}$, $i \in \mathcal{U}$, $j \in \{1, \ldots, d_i\}$. We first show that each individual

symbol sequence has average cost per symbol not exceeding γ. Then we show that superimposing these sequences preserves this property.

Consider the scalar error component $e_{i,j}(t)$, $i \in \mathcal{U}$, $j \in \{1, \ldots, d_i\}$. By (108) we have $|e_{i,j}(kT_i)^+| < L_j$ with strict inequality. So there will be a strictly positive period of time with duration $\tau_{i,j} > 0$ starting at time kT_i until $e_{i,j}(t)$ grows to leave the $[-L_j, L_j]$ box. During this time, no non-free symbols will be transmitted. The "dead time" $\tau_{i,j}$ is simply the amount of time required for the bound $L_j\Big(e^{a_i T_i}(1 + 2e^{T_i}\phi) - 1\Big)/2$ in (106) to grow to size L_j. Specifically, the dead time $\tau_{i,j}$ satisfies $|e_{i,j}(\tau_{i,j} + kT_i)| = L_j$ provided that $|e_{i,j}(kT_i)| \leqslant L_j\Big(e^{a_i T_i}(1+2e^{T_i}\phi)-1\Big)/2$. We now prove that the parameters $\underline{\tau}_i$ were chosen so that

$$|e_{i,j}(\underline{\tau}_i + kT_i)| \leqslant L_j \tag{109}$$

provided that

$$|e_{i,j}(kT_i)^+| \leqslant L_j\Big(e^{a_i T_i}(1 + 2e^{T_i}\phi) - 1\Big)/2, \tag{110}$$

and therefore $\underline{\tau}_i$ lower-bounds the dead time $\tau_{i,j}$. Following a similar process to (95), we have

$$\begin{aligned}
|e_{i,j}(\underline{\tau}_i + kT_i)| &= |\mathbf{v}'_{i,j} e_i(\underline{\tau}_i + kT_i)| && (111)\\
&= |\mathbf{v}'_{i,j} e^{a_i \underline{\tau}_i} G_i(\underline{\tau}_i) e_i(kT_i)^+| && (112)\\
&\leqslant e^{a_i \underline{\tau}_i} \left| \sum_{l=0}^{d_i - j} \frac{\underline{\tau}_i^l}{l!} e_{i,j+l}(kT_i)^+ \right| && (113)\\
&\leqslant e^{a_i \underline{\tau}_i} \Big(|e_{i,j}(kT_i)^+| \\
&\qquad + \sum_{l=1}^{d_i - j} \frac{\underline{\tau}_i^l}{l!} \sum_{l=1}^{d_i - j} |e_{i,j+l}(kT_i)|^+ \Big) && (114)\\
&\leqslant e^{a_i \underline{\tau}_i} \Big(L_j \frac{e^{a_i T_i}(1 + 2e^{T_i}\phi) - 1}{2} \\
&\qquad + \sum_{l=1}^{d_i - j} \frac{\underline{\tau}_i^l}{l!} \sum_{l=1}^{d_i - j} L_{j+l} \Big) && (115)\\
&\leqslant e^{a_i \underline{\tau}_i} L_j \left(\frac{e^{a_i T_i}(1 + 2e^{T_i}\phi) - 1}{2} + e^{\underline{\tau}_i}\phi \right) && (116)\\
&\leqslant L_j, && (117)
\end{aligned}$$

where $\mathbf{v}_{i,j} \in \mathbb{R}^{d_i}$ is a unit vector satisfying (111), (112) follows from the error dynamics (49) in Lemma 4, (113) follows from the definition of the matrix $G_i(\underline{\tau}_i)$, (114) follows from the triangle inequality, (115) follows from the premise (110) and also (108), (116) follows by the definition of ϕ, and by upper-bounding the sum $\sum_{l=1}^{d_i - j} \underline{\tau}_i^l / l!$ by $e^{\underline{\tau}_i}$, and (117) follows from (89). We conclude that $\underline{\tau}_i \leqslant \tau_{i,j}$.

Therefore by (87) we have

$$\tau_{i,j} \geqslant \underline{\tau}_i \coloneqq \frac{1}{a_i} \ln\left(\frac{2}{e^{(a_i+\eta)T_i} - 1}\right) \tag{118}$$

$$= \left(\frac{a_i + \eta}{a_i}\right)\left(\frac{T_i}{h((a_i+\eta)T_i)}\right) \tag{119}$$

$$\Leftrightarrow \qquad \frac{T_i}{\tau_{i,j}} \leqslant \frac{a_i}{a_i+\eta}\gamma < \gamma, \tag{120}$$

where (119) and (120) follow from the definitions of h and T_i. This establishes a bound on the number of non-free transmissions as follows. Consider the symbol sequence $\{s_{i,j}(k)\}_{k\in\mathbb{N}_{>0}}$ emitted by this encoding scheme. Let N_2, N_1 be arbitrary positive integers, and let $N_{\text{nf}} \coloneqq \sum_{k=N_1}^{N_1+N_2-1} I_{s_{i,j}(k)\neq 0}$ be the number of non-free symbols among symbols $s_{i,j}(N_1), \ldots, s_{i,j}(N_1+N_2-1)$. Let $t_l, l \in \{1, \ldots, N_{\text{nf}}\}$ be the time that the lth non-free transmission occurred. The t_l satisfy $N_1 T_i \leqslant t_1 < \ldots < t_{N_{\text{nf}}} \leqslant (N_1+N_2-1)T_i$. Only free symbols are transmitted in the time interval $[t_l, t_l + \tau_{i,j})$, and so

$$t_l \geqslant \tau_{i,j} + t_{l-1}, \qquad \forall l = 2, \ldots, N_{\text{nf}}. \tag{121}$$

Iterating this formula over l, we obtain

$$t_{N_{\text{nf}}} \geqslant \tau_{i,j}(N_{\text{nf}} - 1) + t_1. \tag{122}$$

Rearranging this and using the facts that $N_1 T_i \leqslant t_1$ and $t_{N_{\text{nf}}} \leqslant (N_1+N_2-1)T_i$, we obtain

$$\sum_{k=N_1}^{N_1+N_2-1} I_{s_{i,k}\neq 0} =: N_{\text{nf}} \leqslant \frac{T_i}{\tau_{i,j}} N_2 + 1 \leqslant \gamma N_2 + 1,$$

where we leveraged (120). This implies the average cost per symbol condition (3), so we conclude that for any $i \in \mathcal{U}$ and any $j \in \{1, \ldots, d_i\}$, the symbol sequence $\{s_{i,j}(k)\}_{k\in\mathbb{N}_{>0}}$ has average cost per symbol not exceeding γ.

Finally we show that superimposing the symbol streams results in a stream with average cost per symbol not exceeding γ. Let $N_1, N_2 \in \mathbb{N}$ be arbitrary positive integers, and let $\mathcal{J}_i, i \in \mathcal{U}$ partition $\{N_1, N_1+1, \ldots, N_1+N_2-1\}$ such that $\mathcal{J}_i$ is the set of indices between N_1 and N_1+N_2-1 where the transmitted symbol was sent by sub-encoder i. Then $\sum_{i\in\mathcal{U}} |\mathcal{J}_i| = N_2$, and we obtain

$$\begin{aligned}\sum_{k=N_1}^{N_1+N_2-1} I_{s_{i,k}\neq 0} &= \sum_{i\in\mathcal{U}} \sum_{k\in\mathcal{J}_i} I_{s_{i,k}\neq 0} \\ &\leqslant \sum_{i\in\mathcal{U}} (\gamma|\mathcal{J}_i| + N_{0,i}) \\ &= \gamma N_2 + N_0,\end{aligned}$$

where $N_0 \coloneqq \sum_{i\in\mathcal{U}} N_{0,i}$. The inequality comes from leveraging (3) for each sub-encoder on its respective index interval $\mathcal{J}_i$. This completes the proof of Theorem 3. ∎

5.2 Numerical example

In this subsection we present a numerical example of the event-based encoding scheme from Section 5.

Consider process (1) with

$$A \coloneqq \begin{bmatrix} 57 & -25 \\ 125 & -53 \end{bmatrix} \quad B \coloneqq \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad K \coloneqq \begin{bmatrix} -7 \\ 3.784 \end{bmatrix},$$

for which $\lambda[A] = 2 \pm 10i$ and K is the state-feedback gain of a stabilizing emulation-based controller. Suppose the initial condition is known to lie in the box $\mathcal{X}_0 \coloneqq \{(x_1, x_2) : -1 \leqslant x_i \leqslant 2\}$, and that $x(0) \coloneqq (1, -1)$. Using the coordinate transformation from Lemma 4 yields the open-loop error system $\dot{e}_i(t) = 2e_i(t)$ for $i \in \{1, 2\}$. Note that although the two error components grow at the same rate, their initial conditions are different: $e_1(0) = -3$, $e_2(0) = 2$.

With average bit-rate $r \coloneqq 10$, average cost per symbol $\gamma = 0.2$, and alphabet $\mathcal{A} \coloneqq \{0, 1\}$, the sufficient bound (45) is satisfied. Following the encoder design in Subsection 4.1.1, we pick $N \coloneqq 10$, $M \coloneqq 2$, and $T_i = 1.9$ for $i \in \{1, 2\}$. There are $L(N, M, S) = 56$ length-10 codewords with 2 or fewer non-free symbols. In accordance with the encoder design in Subsection 4.1.1, at time kT_i, $k \in \mathbb{N}_{>0}$, sub-encoder i measures the scalar $e_i(kT_i)$, quantizes it into one of 56 bins — one per codeword — and transmits the appropriate 10-symbol codeword to the decoder. Then the encoder and decoder each update their state estimate according to (57). One observes the state $x(t)$ of the closed-loop system converging to 0. Plots were omitted for space reasons.

Next we demonstrate an event-based controller to stabilize the same system. Note that $r = 10$ and $\gamma = 0.2$ do not satisfy the sufficient bound (82), so they cannot be used in Theorem 3. Instead we use $r \coloneqq 21$, leaving $\gamma \coloneqq 0.2$ as before. This satisfies (84) with $\epsilon = 0.1$, so we apply Theorem 3 to obtain an encoder/decoder and controller that together bound the system $\overline{x}(t) \coloneqq e^{0.1t}x(t)$, and therefore $x(t)$ decays exponentially. This is illustrated in Figures 5 and 6.

With the codeword-based encoder, the two sub-encoders each transmit up to 2 non-free symbols every 1.9 time units, resulting in a total average rate of resource consumption of 2.1 non-free transmissions per time unit. On the other hand, the event-based encoder's two sub-encoders each transmit a symbol every 0.151 time units, and a fraction $\gamma = 0.2$ of these symbols are non-free. Therefore this event-based encoder consumes communication resources at a total average rate of 2.65 non-free transmissions per time unit. This is in accordance with Remark 5: this larger rate of consumption is the price paid for using an easier-to-implement event-based encoding scheme.

6 Conclusion and Future Work

In this paper we considered the problem of bounding the state of a continuous-time linear process under communication constraints. We considered constraints on both the

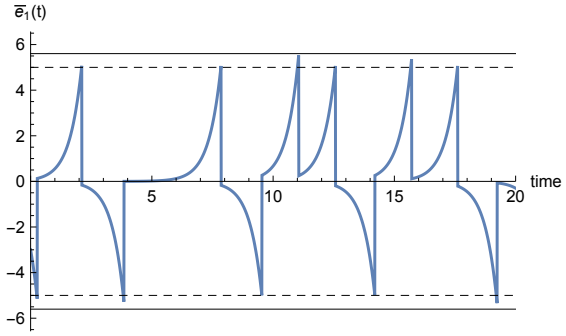

Figure 5: Plot of the closed-loop state estimation error component $\overline{e}_1(t)$ for the $\overline{x}(t)$ system, using the event-based encoding scheme. Once the error leaves $[-L_1, L_1]$ (thin dashed lines), a non-free symbol is transmitted at the next transmission time. The error stays bounded between $-L_1 e^{(a_1+0.1)T_1}$ and $-L_1 e^{(a_1+0.1)T_1}$ (thick dashed lines). Unlike the encoder from Section 4, the transmission of non-free symbols is event-triggered and non-periodic.

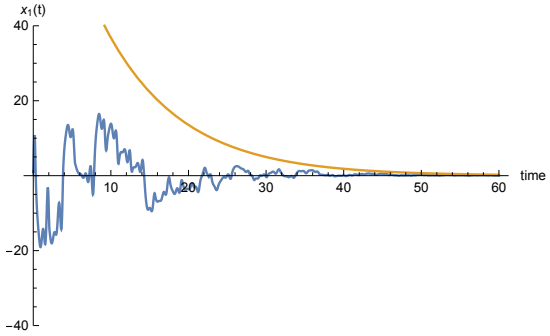

Figure 6: Plot of the closed-loop state $x_1(t)$ exponentially decaying to 0 using the event-based encoding scheme described in Section 5. The curve $100e^{-0.1t}$ is plotted for reference.

channel average bit-rate and the encoding scheme's average cost per symbol. Our main contribution was a necessary and sufficient condition on the process and constraints for which a bounding encoder/decoder/controller exists. In the absence of a limit on the average cost per symbol, the conditions recovered previous work. A surprising corollary to our main result was the observation that one may impose a constraint on the average cost per symbol without necessarily needing to loosen the average bit-rate constraint. Specifically, we proved that if a process may be bounded with a particular average bit-rate, then there exists a (possibly very complex) encoder/decoder that can bound it with that same average bit-rate, while using no more than 50% non-free symbols on average. One would expect that the prohibition of some codewords would require that the encoder necessarily compensate by transmitting at a higher average bit-rate, but this not the case.

Another surprising result was the observation that, for any constraint on average bit-rate and average cost per symbol satisfying the necessary and sufficient conditions for stability, one can always construct a stabilizing encoder with an arbitrarily small

average cost per time unit. In many communication-constrained control problems this is the quantity of interest. We observed that constructing such an encoder boils down to either having precisely-synchronized clocks between the encoder and decoder, or storing a large symbol library on the encoder and decoder.

We then examined an event-based controller and proved its average bit-rate requirements were order-optimal with respect to the necessary and sufficient condition for stabilizability. This supports the use of event-based controllers in limited-communication control schemes.

The controller in the event-based scheme of Section 5 required state feedback. This could be extended to an output-feedback setting by embedding a state observer in the encoder, which is the subject of future work.

A Appendix

Proof of Lemma 1. Let $\ell \in \mathbb{N}_{\geqslant 0}$ be arbitrary. Since the pair's average cost per symbol is at most γ, (3) holds for some $N_0 \in \mathbb{N}_{>0}$. Rearranging (3) yields

$$\sum_{k=N_1}^{N_1+N_2-1} I_{s_k \neq 0} \leqslant N_2\gamma + N_0, \quad \forall N_1, N_2 \in \mathbb{N}_{>0}. \tag{123}$$

Let N be any positive integer greater than $(N_0+1)/\epsilon\gamma$ and define $M \coloneqq \lfloor N\gamma + N_0 + 1 \rfloor$. Invoking (123) for $N_1 \coloneqq \ell N + 1$ and $N_2 \coloneqq N$ yields

$$\begin{aligned} \sum_{k=\ell N+1}^{\ell N+N} I_{s_k \neq 0} &\leqslant N\gamma + N_0 \leqslant M \\ &\leqslant N\gamma + N_0 + 1 < N\gamma(1+\epsilon). \end{aligned} \tag{124}$$

Therefore we have found an M and N satisfying $M < N\gamma(1+\epsilon)$ and moreover (124) implies the condition (8) defining M-of-N encoders. This completes the proof. ∎

Proof of Lemma 4. There exists a real invertible matrix $Q \in \mathbb{R}^{n\times n}$ that transforms A to its real Jordan normal form, namely

$$Q^{-1}AQ = \Lambda \coloneqq \mathbf{diag}\left(J_1, \ldots, J_{n_b}\right),$$

where the J_i are real Jordan blocks: for real eigenvalue a_i with geometric multiplicity d_i, the corresponding real Jordan block $J_i \in \mathbb{R}^{d_i \times d_i}$ has the form

$$\begin{bmatrix} a_i & 1 & \\ & \ddots & \\ & & a_i \end{bmatrix}; \tag{125}$$

for a complex conjuguate pair of eigenvalues $a_i \pm jb_i$ with multiplicity d_i, the associated real Jordan block $J_i \in \mathbb{R}^{2d_i \times 2d_i}$ has the form

$$\begin{bmatrix} \Lambda_i & I_2 & & \\ & \ddots & & \\ & & & \Lambda_i \end{bmatrix}, \tag{126}$$

where the 2-by-2 matrix $\Lambda_i \in \mathbb{R}^{2\times 2}$ has the form

$$\Lambda_i \coloneqq \begin{bmatrix} a_i & b_i \\ -b_i & a_i \end{bmatrix}. \tag{127}$$

Next, define the time-varying invertible block-diagonal matrix $R(t) \in \mathbb{R}^{n\times n}$, $t \geqslant 0$ as

$$R(t) \coloneqq \mathbf{diag}\left(R_1(t), \ldots, R_{n_b}(t)\right) \tag{128}$$

where $R_i(t) \coloneqq I_{d_i} \in \mathbb{R}^{d_i}$ if J_i corresponds to a real eigenvalue a_i, and $R_i(t) \coloneqq \mathbf{diag}(\Theta_i(t)^{-1}) \in \mathbb{R}^{2d_i \times 2d_i}$ if J_i corresponds to a complex conjugate eigenvalue $a_i \pm jb_i$, where

$$\Theta_i(t) \coloneqq \begin{bmatrix} \cos(b_i t) & -\sin(b_i t) \\ \sin(b_i t) & \cos(b_i t) \end{bmatrix} \in \mathbb{R}^{2\times 2}. \tag{129}$$

Let $P(t) \coloneqq R(t)Q^{-1}$, $t \geqslant 0$. We have

$$\begin{aligned} e(t) &\coloneqq P(t)(x(t) - \hat{x}(t)) && (130)\\ &= R(t)Q^{-1}e^{At}(x(0) - \hat{x}(0)) && (131)\\ &= R(t)Q^{-1}e^{Q\mathbf{diag}(J_i)Q^{-1}t}(x(0) - \hat{x}(0)) && (132)\\ &= R(t)e^{\mathbf{diag}(J_i)t}Q^{-1}(x(0) - \hat{x}(0)) && (133)\\ &= R(t)e^{\mathbf{diag}(J_i)t}e(0) && (134)\\ &= R(t)\mathbf{diag}(e^{J_i t})e(0), && (135) \end{aligned}$$

where (133) follows from a well-known property of the matrix exponential, and (134) follows the definition of $e(0)$ and the observation that $R(0)$ is the identity matrix. A well-known property of real Jordan blocks is that

$$e^{J_i t} = e^{a_i t}\begin{bmatrix} 1 & t & \frac{t^2}{2!} & \cdots & \frac{t^{d_i-1}}{(d_i-1)!} \\ & 1 & t & & \\ & & \ddots & & \\ & & & & \\ & & & & 1 \end{bmatrix} \tag{136}$$

if the real Jordan block J_i corresponds to a real eigenvalue, and

$$e^{J_i t} = e^{a_i t} \begin{bmatrix} \Theta_i(t) & \Theta_i(t)t & \Theta_i(t)\frac{t^2}{2!} & \dots & \Theta_i(t)\frac{t^{d_i-1}}{(d_i-1)!} \\ & \Theta_i(t) & \Theta_i(t)t & & \\ & & \ddots & & \\ & & & & \\ & & & & \Theta_i(t) \end{bmatrix} \tag{137}$$

if it corresponds to a complex conjugate pair. In terms of $R_i(t)$ and $G_i(t)$ these equations become simply

$$e^{J_i t} = e^{a_i t} R_i(t)^{-1} G_i(t). \tag{138}$$

Using this in (135) yields

$$e(t) = R(t)\mathbf{diag}(e^{a_i t}\mathbf{diag}(R_i(t)^{-1}G_i(t)))e(0) \tag{139}$$

$$e(t) = \mathbf{diag}(R_i(t))\mathbf{diag}(R_i(t)^{-1})\mathbf{diag}(e^{a_i t}G_i(t))e(0) \tag{140}$$

$$e(t) = \mathbf{diag}(e^{a_i t}G_i(t))e(0), \tag{141}$$

implying (49).
Lastly, it is straightforward to verify that the minimum singular value of $R_i(t)$ is $\sigma_{\min}(R_i(t)) = 1$ for any t. Moreover, since Q is invertible, there exists $\epsilon > 0$ for which $\sigma_{\min}(P(t)) \geqslant \epsilon$ for all t. This concludes the proof. ∎

Lemma 5. *For any $N \in \mathbb{N}_{>0}$ and $M \in \mathbb{R}_{\geqslant 0}$ with $M \leqslant N$, every M-of-N encoder has average cost per symbol not exceeding M/N.*

Proof of Lemma 5. Suppose M and N are fixed and consider a sequence of N_2 symbols starting at index N_1, for arbitrary $N_1, N_2 \in \mathbb{N}_{>0}$. This index sequence $\{N_1, \dots, N_1 + N_2 - 1\}$ overlaps or partially overlaps with at most $\lceil N_2/N \rceil + 1$ of the fixed N-symbol codewords. Each codeword has at most M non-free symbols. Therefore the number of non-free symbols in the sequence is upper-bounded by

$$\begin{aligned} \sum_{k=N_1}^{N_1+N_2-1} I_{s_k \neq 0} &\leqslant M\left(\lceil N_2/N \rceil + 1\right) \\ &\leqslant M(N_2/N + 2) = \frac{M}{N}N_2 + 2M. \end{aligned} \tag{142}$$

We let $N_0 \coloneqq 2M$ and rearrange terms to obtain (3), the definition of average cost, with $\gamma = M/N$. ∎

Lemma 6. *The following inequality holds for all $N, S \in \mathbb{N}_{>0}$, $q \in (0, S/(S+1)]$, and $i \in [0, Nq]$:*

$$q^i(1-q)^{N-i} \geqslant 2^{-N\,H(q)}\,\frac{S^i}{S^{Nq}} \tag{143}$$

where $H(q) \coloneqq -q\log_2 q - (1-q)\log_2(1-q)$ is the base-2 entropy of a Bernoulli random variable with parameter q.

Proof of Lemma 6. Let N, S, q, and i take arbitrary values from the sets described in the lemma's statement. Since $\log_2$ is a monotone increasing function, $\log_2(q/(1-q))$ for $q > 0$ is maximized at the right endpoint value, $q = S/(S+1)$, where it equals $\log_2 S$. This leads to

$$\log_2 q - \log_2(1-q) \leqslant \log_2 S \tag{144}$$

for all $S \in \mathbb{N}_{>0}$ and $q \in (0, S/(S+1)]$. Next, $i \in [0, Nq]$ by assumption, therefore $i - Nq \leqslant 0$. Multiplying (144) by $i - Nq$ and straightforward algebraic manipulation yields

$$\begin{aligned} & i \log_2 q + (N-i)\log_2(1-q) \\ & \geqslant Nq \log_2 q + N(1-q)\log_2(1-q) + (i-Nq)\log_2 S \\ & = -NH(q) + (i-Nq)\log_2 S, \end{aligned}$$

where the equality follows from the definition of $H(q)$. Raising 2 to the power of both sides, (143) follows. ∎

References

[1] K. Åström. Event based control. In *Analysis and Design of Nonlinear Control Systems: In Honor of Alberto Isidori*, page 127. Springer Verlag, 2007. (cited in p.)

[2] K. Åström and B. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 2, pages 2011 – 2016, dec. 2002. doi: 10.1109/CDC.2002.1184824. (cited in p.)

[3] R. Brockett and D. Liberzon. Quantized feedback stabilization of linear systems. *Automatic Control, IEEE Transactions on*, 45(7):1279–1289, Jul 2000. ISSN 0018-9286. doi: 10.1109/9.867021. (cited in p.)

[4] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 2012. ISBN 9781118585771. (cited in p.)

[5] R. Goebel, R. Sanfelice, and A. Teel. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, 2012. ISBN 9780691153896. (cited in p.)

[6] J. P. Hespanha, A. Ortega, and L. Vasudevan. Towards the control of linear systems with minimum bit-rate. In *Proc. of the Int. Symp. on the Mathematical Theory of Networks and Syst.*, Aug. 2002. (cited in p.)

[7] E. Kofman and J. Braslavsky. Level crossing sampling in feedback stabilization under data-rate constraints. In *Decision and Control, 2006 45th IEEE Conference on*, pages 4423–4428, Dec 2006. doi: 10.1109/CDC.2006.377483. (cited in p.)

[8] D. Lehmann and J. Lunze. Event-based control using quantized state information. In *Proc. of the 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'10)*, pages 1–6, Sep. 2010. (cited in p.)

[9] L. Li, X. Wang, and M. Lemmon. Stabilizing bit-rates in quantized event triggered control systems. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pages 245–254. ACM, 2012. (cited in p.)

[10] J. Lunze and D. Lehmann. A state-feedback approach to event-based control. *Automatica*, 46(1):211 – 215, 2010. ISSN 0005-1098. doi: http://dx.doi.org/10.1016/j.automatica.2009.10.035. (cited in p.)

[11] A. Matveev and A. Savkin. Multirate stabilization of linear multiple sensor systems via limited capacity communication channels. *SIAM Journal on Control and Optimization*, 44(2):584–617, 2005. doi: 10.1137/S0363012902419965. (cited in p.)

[12] A. Matveev and A. Savkin. An analogue of Shannon information theory for detection and stabilization via noisy discrete communication channels. *SIAM Journal on Control and Optimization*, 46(4):1323–1367, 2007. doi: 10.1137/040621697. (cited in p.)

[13] G. Nair. A nonstochastic information theory for communication and state estimation. *Automatic Control, IEEE Transactions on*, 58(6):1497–1510, June 2013. ISSN 0018-9286. doi: 10.1109/TAC.2013.2241491. (cited in p.)

[14] G. Nair and R. Evans. Communication-limited stabilization of linear systems. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 1, pages 1005 –1010, 2000. doi: 10.1109/CDC.2000.912906. (cited in p.)

[15] G. N. Nair and R. J. Evans. Exponential stabilisability of finite-dimensional linear systems with limited data rates. *Automatica*, 39(4):585 – 593, 2003. ISSN 0005-1098. doi: http://dx.doi.org/10.1016/S0005-1098(02)00285-6. (cited in p.)

[16] J. Pearson, J. P. Hespanha, and D. Liberzon. Control with minimum communication cost per symbol. In *Proc. of the 53nd Conf. on Decision and Contr.*, pages 6050–6055, Dec. 2014. (cited in p.)

[17] J. Pearson, J. P. Hespanha, and D. Liberzon. Quasi-optimality of event-based encoders. In *Proc. of the 54nd Conf. on Decision and Contr.*, pages 4800–4805, Dec. 2015. doi: 10.1109/CDC.2015.7402968. (cited in p.)

[18] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *Automatic Control, IEEE Transactions on*, 52(9):1680 –1685, sept. 2007. ISSN 0018-9286. doi: 10.1109/TAC.2007.904277. (cited in p.)

[19] P. Tallapragada and N. Chopra. On co-design of event trigger and quantizer for emulation based control. In *American Control Conference (ACC), 2012*, pages 3772–3777, June 2012. doi: 10.1109/ACC.2012.6315501. (cited in p.)

[20] P. Tallapragada and J. Cortes. Event-triggered stabilization of linear systems under bounded bit rates. *ArXiv e-prints*, May 2014. (cited in p.)

[21] S. Tatikonda and S. Mitter. Control under communication constraints. *Automatic Control, IEEE Transactions on*, 49(7):1056 – 1068, july 2004. ISSN 0018-9286. doi: 10.1109/TAC.2004.831187. (cited in p.)

[22] S. C. Tatikonda. *Control under communication constraints*. PhD thesis, Massachusetts Institute of Technology, 2000. (cited in p.)